SunVTS™ 6.2 Test Reference Manual for SPARC® Platforms

Sun Microsystems, Inc.
www.sun.com

Part No. 819-6455-10
June 2006, Revision A

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

Contents

Preface

SunVTS™ is the Sun Microsystems™ Inc. Validation Test Suite. SunVTS is a comprehensive software diagnostic package that tests and validates Sun SPARC® and x86 hardware by verifying the configuration and functionality of controllers, devices, and platforms.

SunVTS is primarily used from a graphical user interface (GUI). This document describes SunVTS tests that run on machines with SPARC architectures. The descriptions include specific test options, procedures, and error messages.

This document is a reference for SunVTS test-specific information. Refer to the *SunVTS User's Guide* for overall SunVTS information.
Developers or experienced users who want to run the SunVTS diagnostic application will find these documents useful.

Note – The Solaris™ release with which this version of SunVTS is delivered supports systems that use the SPARC and x86 families of processor architectures UltraSPARC®, SPARC 64, AMD64, Pentium, and Xeon EM64T. The supported systems appear in the *Solaris 10 Hardware Compatibility List* at `http://www.sun.com/bigadmin/hcl`. This document cites any implementation differences between the platform types.

Before You Read This Book

To fully use the information in this document, you must have thorough knowledge of the topics described in these documents:

- *SunVTS User's Guide*
- *SunVTS Quick Reference Card*

How This Book Is Organized

This book is organized as follows:

Chapter 1 describes SunVTS requirements, test modes, user interfaces, the collection of tests, and how to run a test from the command line.

The remaining chapters describe the individual SunVTS tests, their options, applicable test modes, and command-line syntax. These chapters are arranged in alphabetical order according to test name.

Appendix A provides information about the serial and parallel port loopback connectors that are required by some of the SunVTS tests.

Using UNIX Commands

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

Refer to one or more of the following for this information:

- *Solaris Handbook for Sun Peripherals*
- Software documentation that you received with your system
- Solaris operating system documentation, which is at:

  `http://docs.sun.com`

Shell Prompts

Shell	Prompt
C shell	*machine-name*`%`

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

Typographic Conventions

Typeface*	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`
`AaBbCc123`	What you type, when contrasted with on-screen computer output	`%` **`su`** `Password:`
AaBbCc123	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be superuser to do this. To delete a file, type `rm` *filename*.

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

Related Documentation

This document covers SunVTS version 6.2 tests. It serves as a reference companion to the SunVTS documents listed below.

Application	Title	Part Number
Installation and Navigation	*SunVTS 6.2 User's Guide*	819-2361-10
Quick Reference Card	*SunVTS Quick Reference Card*	819-2365-10

Accessing Sun Documentation

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

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

Third-Party Web Sites

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

Contacting Sun Technical Support

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

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

Sun Welcomes Your Comments

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

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

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

SunVTS 6.2 Test Reference Manual for SPARC Platforms, part number 819-6455-10

CHAPTER 1

SunVTS Overview

This chapter contains the following topics:

The Sun Validation and Test Suite (SunVTS) software performs multiple diagnostic hardware tests from a single user interface. SunVTS verifies the connectivity, functionality, and reliability of controllers and devices.

Use SunVTS to test one device or multiple devices. Some of the major test categories are as follows:

- Audio tests
- Communication (serial and parallel) tests
- Graphic or video tests
- Memory tests
- Network tests
- Peripherals (disks, tape, CD-ROM, DVD-ROM, printer, diskette) tests
- Processor tests
- Storage tests

SunVTS comprises of many individual tests that support testing of a wide range of products and peripherals. Most of the tests can test devices in a 32-bit or 64-bit Solaris operating system (OS).

Such flexibility requires that the proper test modes and options need to be selected to maximize its effectiveness. This document covers the individual test options, modes, and requirements. For overall test configuration modes and options refer to the *SunVTS User's Guide*.

Note – When an error occurs in SunVTS testing, the test message window displays the error number, the error description, the probable cause of the error, and the recommended actions. Because this information is displayed at the time of the error, error messages are not included in this document.

The default installation directory for SunVTS is `/opt/SUNWvts`. However, when you are installing SunVTS, you can specify a different directory. Refer to the *SunVTS User's Guide* for installation information.

Note – SunVTS does not support processor sets. If processor sets are defined, you must first delete the processor sets before running SunVTS.

SunVTS Version Information

The standard command line argument, `-V`, displays the SunVTS version and release date of the test, if available.

SunPCi-3 Card Support

`sunpci2test` now supports the SunPCi™ III cards. Solaris 10 OS supports SunPCi-III Version 3.2.2 with Patch 118591-03 only. Solaris 10 does not support the SunPCi-2 card.

Test Requirements

SunVTS 6.2 was first introduced and designed to run in the Solaris 10 1/06 (Solaris 10 Update 1) OS and subsequent compatible releases. SunVTS 6.2 is not supported on OS releases prior to Solaris 10 3/05 (Solaris 10).

The operating system kernel must be configured to support all peripherals that are to be tested.

Some SunVTS tests have special requirements such as the connection of loopback connectors, installation of test media, or the availability of disk space. These requirements are listed for each test in the corresponding chapter in this document.

Collection of SunVTS Tests

Many individual tests make up the SunVTS collection of tests. Each test is a separate process from the SunVTS kernel. Each test can be run individually from the command line or from the SunVTS user interface.

When SunVTS is started, the SunVTS kernel automatically probes the system kernel to determine the hardware devices. The devices are then displayed on the SunVTS control panel with the appropriate tests and test options. This provides a quick check of your hardware configuration, and no time is wasted trying to run tests that are not applicable to your configuration.

During testing, the hardware tests send the test status and messages to the SunVTS kernel through interprocess communication (IPC) protocols. The kernel passes the status to the user interface and logs the messages.

SunVTS has a shared object library that contains test-specific probing routines. At runtime, the SunVTS kernel dynamically links in and calls these probing routines to initialize its data structure with test-specific information. You can add new tests into the SunVTS environment without recompiling the SunVTS source code.

Beginning with SunVTS 3.0, the SunVTS kernel and most tests support 32-bit and 64-bit operating systems. When the `sunvts` command is used to start SunVTS, the appropriate tests (32-bit or 64-bit versions) are presented.

32- and 64-Bit Tests

In Solaris 10 or later OSs, only 64-bit compatible tests are supported. Because each test is a separate program, you can run individual tests directly from the command line. Run tests from specific directories as follows:

- 32-bit tests – `/opt/SUNWvts/bin/`*testname*
- 64-bit tests – `/opt/SUNWvts/bin/sparcv9/`*testname*

If you are not sure which OS is running, refer to the Solaris System Administration manuals. In Solaris 10 OS, you can use the following command to identify the application support of your system.

```
# isainfo -v
```

Note – The `isainfo` command is not available in Solaris 2.6 or earlier releases.

SunVTS User Interfaces

You can run SunVTS tests from the JDS graphical user interface or the TTY interface. SunVTS tests can also be run individually from a shell tool command line, using the command-line syntax for each test (refer to "Running a Test From the Command Line" on page 6). TABLE 1-1 describes the various SunVTS user interfaces. Refer to the *SunVTS User's Guide* for more information on these interfaces.

TABLE 1-1 SunVTS System Interfaces

SunVTS System Interfaces	Description
Graphical user interface (GUI)	Select tests and test options with a mouse in the Solaris Java Desktop System (JDS) interface.
TTY interface	Run SunVTS from a terminal or modem attached to a serial port. You use the keyboard instead of the mouse. The interface displays one screen of information at a time.
Command-line execution	Run each of the SunVTS tests individually from a shell tool using the command-line syntax. Each test description in this document contains the corresponding command-line syntax.

Running a Test From a User Interface

The common way to run SunVTS testing is through a SunVTS user interface—JDS or the TTY interface.

Test configuration, control, and results are easily accessed through buttons and dialog boxes. These buttons and dialog boxes are covered in the *SunVTS User's Guide*. However, the Test Parameter Options dialog box is unique for each test, and is therefore covered in this manual.

Test Parameter Options Dialog Box

The options displayed in this menu differ for each test, but the Apply menu, and the Reset and Cancel buttons are generic. TABLE 1-2 describes all the items.

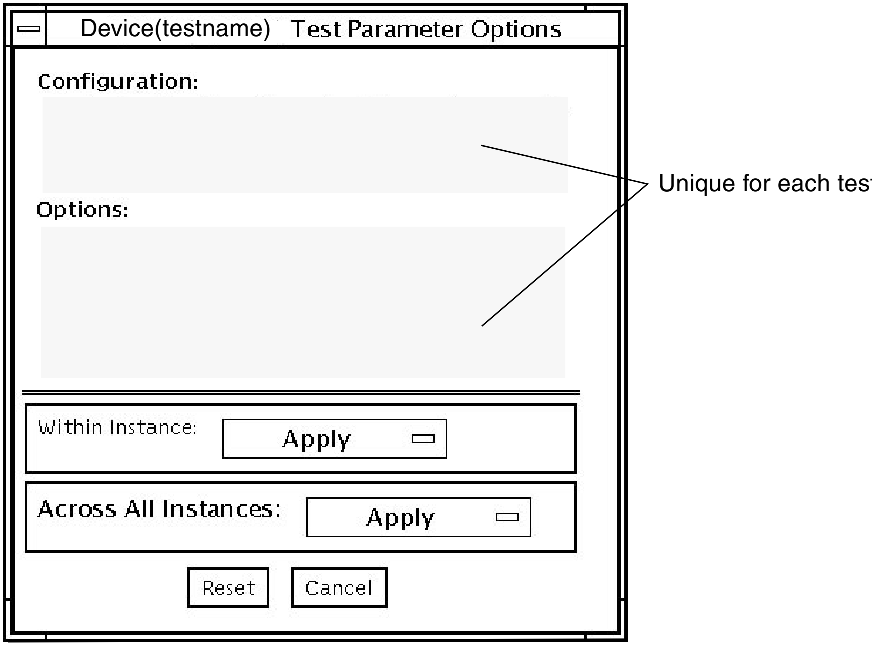


FIGURE 1-1 Test Parameter Options Dialog Box

TABLE 1-2 Test Parameter Options Dialog Box Items

Menu Item	Description
Configuration	Information such as device type, capacity, revision, and serial numbers for the selected device. This information cannot be changed.
Options	A list of test options that are used to customize the testing of the selectable device, group, or all devices. The options are specific for each test and are covered in the test specific-chapters in this manual.
Within Instance	Provides the means to apply the settings: • To this device only with Apply • To all devices within this group with Apply to Group • To all devices (of the *same device type* for *all controllers*) with Apply to All The option settings are only applied to one instance of the test.

TABLE 1-2 Test Parameter Options Dialog Box Items *(Continued)*

Menu Item	Description
Across All Instances	Provides the means to apply the settings globally: • To this device only with Apply • To all devices within this group with Apply to Group • To all devices (of the *same device type* for *all controllers*) with Apply to All The option settings are applied to all instances.
Reset	Returns the option values to their default settings and closes the Test Parameter Option dialog box.
Cancel	Ignores any changes made to option values and closes the Test Parameter Option dialog box.

Note – The Test Parameter Options dialog box descriptions also apply to the Test Parameter Options menu in the TTY interface.

Running a Test From the Command Line

In some cases it may be more convenient to run a single SunVTS test from the command line rather than through a SunVTS user interface. The following information describes how to do this.

Unless specified, the test runs without the SunVTS kernel (`vtsk`). All events and errors are sent to `stdout` or `stderr` and are not logged in the log files.

When you run a test in this way, you must specify all test options in the form of command-line arguments. There are two types of command-line arguments:

- Standard arguments—common to all tests. Refer to TABLE 1-3 for details.
- Test specific arguments—unique to a specific test. Refer to the test-specific chapters in this book for details.

The standard syntax for all SunVTS tests is:

`testname` `[`**`-scruvdtelnf`**`]` `[`**`-i`** *number*`]` `[`**`-w`** *number*`]` `[`**`-o`** *test-specific-arguments*`]`

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 might be available as a 32-bit test only. For more information, see "32- and 64-Bit Tests" on page 3.

Standard Command-Line Arguments

The following table defines the standard SunVTS command-line arguments:

TABLE 1-3 Standard Command-Line Arguments

Argument	Description
`-s`	Runs a test as though it were invoked from the SunVTS kernel (`vtsk`). The default is to send the output to `stdout` or `stderr`.
`-c`	Enables a core image of the test process to be created in the current working directory upon receipt of certain signals, otherwise those signals are caught and handled to prevent a core from being generated. The default is to disable the creation of a core image.
`-r`	Enables run on error so that when an error occurs, the test continues with the next test sequence instead of exiting. The default is false.
`-v`	Runs the test in Verbose mode and displays messages with more detailed information about the testing process. The default is false.
`-V`	Displays the SunVTS version and release date of the test.
`-d`	Runs the test in Debug mode and displays messages to help programmers debug their test code. The default is false.
`-t`	Runs the test in Trace mode and displays messages that track function calls and sequences currently in use by the test code. The default is false.
`-l`	Runs the test in Online Functional mode. This is the same mode that tests run in when executed with the `vtsui.online` command. It is a non-intrusive version that will not significantly affect other applications. See the note below. The default is true.
`-x`	Runs the test in Exclusive mode.
`-n`	Runs the test in Connection mode. See the note below. The default is false.
`-f`	Runs the test in full Functional mode. This mode assumes that the test has complete control of the device under test. See the note below. The default is false.
`-p` *number*	Defines the number of passes for scalable tests. The default is 1.
`-i` *number*	Defines the number of instances for scalable tests. The default is 1.
`-w` *number*	Defines to which instance the test is assigned; this option is for scalable tests. The default is 0.
`-o`	Indicates that the options and arguments that follow are test specific.

Note – Separate each test-specific argument by commas, with no space after each comma.

Note – If you choose to specify a test mode with the `l`, `n`, or `f` option, specify only one option at a time because only one test mode can be selected at a time.

Test-Specific Arguments

SunVTS includes test-specific arguments that follow the format specified in the `getsubopt(3c)` man page. Separate each test-specific argument by commas, with no space after the comma. For example: `#`**`./sample -v -o dev=/dev/audio,volume=78`**. For information about test-specific arguments refer to the specific test chapter in this document.

Frame Buffer Tests

SunVTS includes a number of tests that exercise frame buffers:

- `cryptotest`
- `graphicstest`
- `ibhcatest`
- `ifbtest` (Expert 3D)
- `jfbtest` (XVR-1200)
- `m64test`
- `pfbtest` (XVR-1000)

If you are testing more than one frame buffer, follow these guidelines and instructions.

Caution – Disable the Power Management screen saver option and the Save/Resume option before you run any of the SunVTS frame buffer tests. For information on disabling these Power Management features, refer to the Power Management chapter in the *Solaris Common Desktop Environment: Users's Guide* in the Solaris 9 User Collection. This document is available at: `docs.sun.com`.

Caution – If you are using the JDS interface for SunVTS, do not conduct frame buffer tests through the `dtlogin` window. Log in as `root` and disable the auto-logout option.

Caution – Do not run TTY mode and frame buffer tests concurrently on the console monitor. The frame buffer test may fail.

Testing Multiple Frame Buffers

The following rules apply when you test multiple frame buffers (displays) simultaneously:

- Only the console monitor can run the window environment (such as JDS). The console monitor is the monitor connected to the frame buffer appointed by `/dev/fb`. SunVTS enables frame buffer locking on the console monitor by default.
- The frame buffer that is running the window environment must have window locking enabled to avoid false test failures. All other frame buffers must have window locking disabled.
- The group concurrency for graphics must be set to 1.

Remote Testing of Frame Buffers

If you start `sunvts` or `vtsk` from a screen other than the console monitor, frame buffer locking is not available. In this case:

- Disable the window locking option on the remote screen by setting it to `d`.
- Enable frame buffer locking for the console monitor, as shown in the example above. The SunVTS user interface cannot display on a monitor if locking is disabled.

Do not run any graphic programs (including `vtsui`) on the remote frame buffer during graphic testing.

Locking Frame Buffers

If you are testing multiple frame buffers or remote frame buffers, you might need to enable or disable frame buffer locking.

▼ To Enable Frame Buffer Locking

- **Take one of the following actions:**
  - In the JDS SunVTS interface, go to the Option menu of the graphics test and select Enable for the Frame Buffer Locking option.
  - At the command line, use the `lock=e` option.

  For example:

```
# ./fbtest -o dev=cgthree0, lock=enable
```

▼ To Disable Frame Buffer Locking

- **Take one of the following actions:**
  - In the JDS SunVTS interface, go to the Option menu of the graphics test and select Disable for the Frame Buffer Locking option.
  - At the command line, use the `lock=d` option. For example:

```
# ./fbtest -o dev=cgthree0, lock=disable
```

CHAPTER 2

SunATM Adapter Test (atmtest)

atmtest Description

The atmtest checks the functionality of the SunATM™ 155 and SunATM 622 SBus and PCI bus adapters.

It runs only in loopback (external or internal) mode. The asynchronous transfer mode (ATM) adapter, and ATM device driver must be present. To run the atmtest in external loopback mode, a loopback connector must be attached to the ATM adapter. The internal loopback mode does not require a loopback connector.

atmtest uses DLPI RAW mode to talk to the device driver. The test establishes a virtual circuit (VC) to send a message, receive a message, and compare messages. If the messages do not match, or if the message is out of sequence, atmtest displays an error message.

Using a random number generator, atmtest sends data into a data buffer and then sends each message from a different starting point. This assures that no two consecutive messages are the same.

atmtest can test more than one virtual circuit. Using more virtual circuits increases the stress level of the test. atmtest automatically selects the virtual circuit number that is unique to the test.

atmtest is nonscalable because it provides multiple virtual circuits to be tested by a single instance.

`atmtest` Test Requirements

`atmtest` can only be selected when the Intervention mode is enabled, since it requires a loopback connector for external loopback testing. While Intervention mode is enabled, `atmtest` and `nettest` are both available as default selections However, you must deselect `nettest` when testing the ATM device.

Shutdown the ATM interface to make sure that the interface is in offline mode before running `atmtest`.

Note – Do not run `nettest` while running `atmtest`.

Note – The external optical loopback test requires a 62.5 micron cable.

`atmtest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

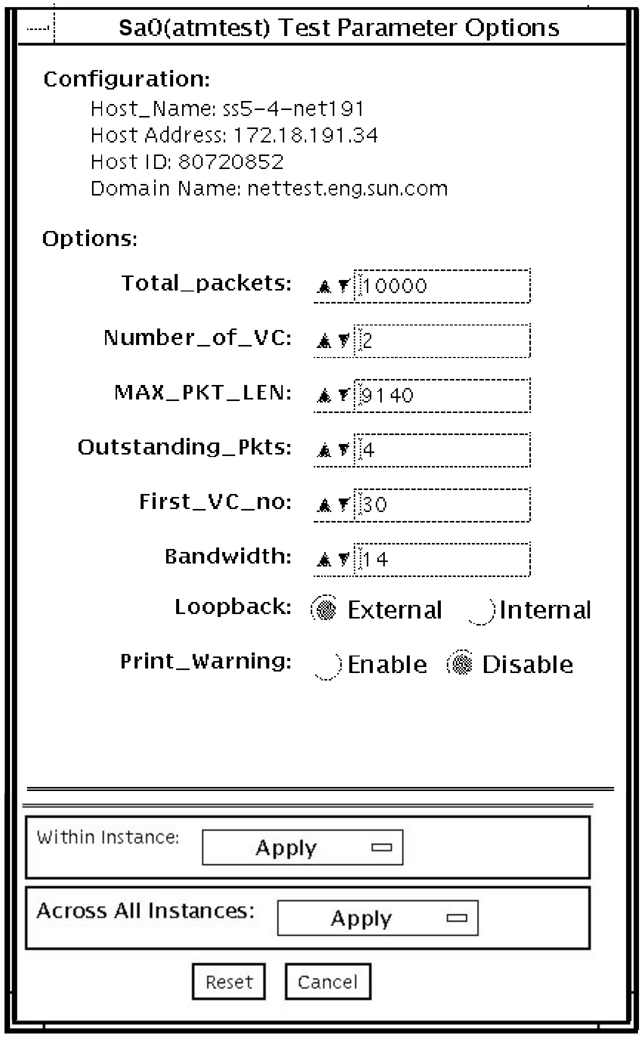


FIGURE 2-1 `atmtest` Test Parameter Options Dialog Box

TABLE 2-1 `atmtest` Options

Options	Description
Configuration	The post address, host ID, and domain name of the system being tested.
Total packets	The total number of packets sent. The default number of packets sent is 10000.
Number of VC	The default number of virtual circuits is 2. The `atmtest` uses these two virtual circuits to send out messages simultaneously. The message is received in sending order.
MAX-PKG-LEN	The maximum packet length to be used by the test to send out the data. The default number is 9140.
Outstanding-pkts	The maximum number of outstanding packets. `atmtest` stops sending messages when the outstanding packet count is more than the number of packets this field specifies.
First-VC-no	Enables the user to set up the starting virtual circuit number to be used for each `atmtest` instance. `atmtest` can automatically avoid virtual circuit numbers that have already been used.
Bandwidth	Enables the user to select different bandwidths to test. The default number is 14.
Loopback	Enables the user to select either the external loopback field or internal loopback field. The default selection is the external loopback field. A loopback connector is only needed for external loopback testing.
Print Warning	Disabled by default. Click Enable to see warning errors, such as retry on timeout.

atmtest Test Modes

TABLE 2-2 atmtest Supported Test Modes

Test Mode	Description
Functional	Runs the full set of tests.

atmtest Command-Line Syntax

/opt/SUNWvts/bin/atmtest *standard-arguments*,**-o dev=***device*,**tpkts=***n*,**nv=***n*,**ml=***n*,**bw=***n*,**opkts=***n* | **warn**|**ld**|**sd**|**sl**|**nc**|**ns**,**vcf**=*n*

TABLE 2-3 atmtest Command-Line Syntax

Argument	Description
dev=*device*	Specifies the device name to be tested, such as ba0 or sa0.
tpkts=*n*	Specifies the number of packets to loopback, -1 for continuous [1...2147483647, -1].
nv=*num-vc*	Specifies the number of simultaneous virtual circuits to be tested.
ml=*max-len*	Specifies the maximum length of the random packet.
bw=*bandwidth*	Specifies the bandwidth in Mbyte/s of a virtual circuit.
opkts=*n*	Specifies the number of packets for each virtual circuit that can be transmitted without receiving a corresponding packet.
warn	When enabled, prints warning messages.
ld	The internal loopback mode is selected.
sd	Changes the payload data to static instead of random.
sl	Changes all packets to their maximum length.
nc	Instructs the test not to check the receive payload (improves throughput).
ns	Instructs the test not to exit on a packet reception failure.
vcf=*n*	Specifies the first virtual circuit number used.

CHAPTER 3

Audio Test (audiotest)

audiotest Description

The audiotest verifies the hardware and software components of the audio subsystem. This test supports all Sun audio implementations.

Note – audiocs and audio drivers could cause hangs when performing audiotest with other SunVTS tests during very heavy system load. audio driver hangs could occur on audbri platforms such as SS10. audiotest captures the hangs caused by the audiocs driver and occasionally attempts to retry.

This test works with exclusive access devices (only one process or application available at a time), or with newer audio devices that support the software mixer feature available in the Solaris OS.

Note – audiotest turns the mixer off automatically at run time. Shut down all audio applications before running audiotest, as Online mode is not supported. The mixer is restored after testing.

This test is not scalable.

The availability of the following subtests depends on the particular audio implementation being tested.

audiotest Subtests

TABLE 3-1 audiotest Subtests

Subtest	Description
Record/Play test	This test plays and records one second of data, but does not check the data. This test runs on all audio implementations.
Crystal test	This test measures the accuracy of the crystal that generates the sample rate clock, by playing a one-second signal and then measuring the actual time required to play the signal. This measurement is performed for each of the eight standard sample rates. This test is available for `dbri`(7) and `audiocs`(7) audio implementations.
Loopback tests	This test verifies the functionality and signal quality of the audio ports. The test simultaneously plays and records a known signal. The recorded signal is analyzed for loop gain and signal-to-noise ratio plus distortion. This analysis is repeated at various sample rates, encodings, precisions, and channels. The audio ports that are supported depend on the audio implementation under test. The `audiocs(7)` implementation supports loopbacks from and to headphone, line-out, microphone, and line-in ports. The `dbri(7)` and speakerbox implementation supports fewer ports. The `audioamd(7)` implementation does not support loopback tests. Most tests require a stereo loopback cable. Note: The microphone loopback tests require special hardware, and are used by manufacturing centers and special test facilities. Do not invoke the microphone loopback tests unless you have the required hardware.
Controls test	This test verifies the three control buttons on the Sun speakerbox. The Controls test plays music while you press the Volume Down, Volume Up, and Mute buttons in a specific order. If no button is pressed in 30 seconds the test fails. This test is only supported on the `dbri`(7) and speakerbox implementation.
Audio test	This test plays a 30-second music file out of the speaker or headphone. The full benefit of this test is only realized if the user listens to the output. Badly distorted audio or inaudible music indicates a problem. This test is supported on all audio implementations.

audiotest Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

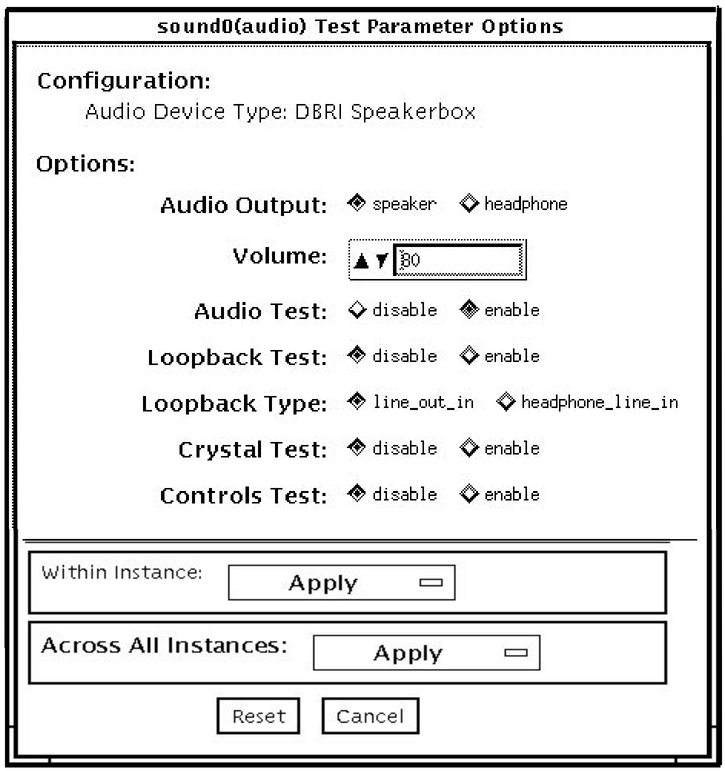


FIGURE 3-1 audiotest Test Parameter Options Dialog Box

Note – Upon startup, the SunVTS probe utility determines which audio implementation is present and adjusts the audiotest Option menu appropriately. Your dialog box might look different than the one pictured here, but will contain some or all of these options.

Note – The internal loopbacks are only active if the audio jacks are unused (nothing is connected).

Some options can only be selected through the command line. See the command-line option descriptions in "audiotest Command-Line Syntax" on page 23.

TABLE 3-2 audiotest Options

Option	Description
Audio Output	Selects the output port for the Music Play test.
Volume	Sets the volume for the Music Play test.
Audio test	Enables or disables the Music Play test. This test is enabled by default.
Loopback test	Enables or disables the Loopback test. A loopback cable must be installed between the selected ports to run external loopback tests. This test is disabled by default.
Loopback type	Selects the type of Loopback test to run.
Crystal test	Enables or disables the Crystal test. This test is disabled by default.
Controls test	Enables or disables the speakerbox Controls test. This is an interactive test. The user is prompted to press the control buttons on the speakerbox. This test is disabled by default.

Note – Do not run the Crystal test while running other SunVTS tests. The Crystal test is timing-dependent. If the system is too busy, it fails due to time-out errors.

`audiotest` Test Modes

TABLE 3-3 `audiotest` Supported Test Modes

Test Mode	Description
Connection test	A simple open and close is performed. No data is transferred. The test returns a pass if the device can be opened and closed successfully. If the device cannot be opened because it is busy, then it is assumed that the device is successfully connected to another process and the test passes.
Functional	The record/play test is run and you can choose to run any of the tests described earlier. In this mode, the test will fail if the device is busy.

`audiotest` Command-Line Syntax

`/opt/SUNWvts/bin/audiotest` *standard_arguments* **`-o dev=/dev/sound/`***unit_no***`,I=/dev`***ioctl_device***`,M,L,Q,S,T=`***loopback_test_type***`,X,E,LE,CD,CDD=`***CD_device_name***`,CDT=`***track_number***`,CDG=`***play_gain***`,CDL=`***play_time***`,W,MF=`***filename***`,TF=`***filename*

TABLE 3-4 `audiotest` Command-Line Syntax

Argument	Description
`dev=/dev/`*audio-device*	Specifies the audio device to be tested. The default is `dev=/dev/audio`.
`I=/dev/`*ioctl-device*	Specifies the `audio ioctl` device to be tested. The default is `/dev/audioctl.`
`M`	Enables the Music Play test.
`L`	Enables the Loopback test.
`Q`	Enables the Quality test. This option does the same thing as `L` option except that it prints an extra status message upon completion.
`S`	Enables the speakerbox Controls test.

TABLE 3-4 `audiotest` Command-Line Syntax *(Continued)*

Argument	Description
T=*loopback_test_type*	Specifies the type of Loopback test. The default is 1. The choices are listed below: • 0—Codec Internal Loopback (CS4231 audio only) • 1—Line-in/Line-out • 2—Headphone/Line-in • 3—Headphone/Microphone • 4—Speaker/CD-input • I1—Internal Line-in/Line-out • I2—Internal Spk/Mic • I3—Internal Headphone/Aux1 • I4—Internal Speaker/Aux1 • I5—Internal Headphone/Mic Note: Test type 0 is always run by default on CS4231 audio implementations. Test types 3 and 4 require special hardware, and are used by manufacturing centers and special test facilities. Do not invoke these tests unless you have the required hardware.
X	Enables the Audio Crystal test.
E	Continues testing if an error occurs.
LE	Loops on error. This plays the signal data in a continuous loop.
CD	Enables the `cdtest`. This is for systems with an internal CD-ROM drive. A CD-ROM with music tracks must be loaded prior to running this test.
CDD=*CD_device_name*	Specifies the raw device name for the CD-ROM drive. The default is `CDD=/dev/rdsk/c0t6d0s0`.
CDT=*number*	Specifies the track number of the CD-ROM to play. The default is to play the first track on the disc.
CDG=*play_gain*	Specifies the play gain of the CD Play test (0 to 255). The default is 120.
CDL=*play_time*	Specifies the number of seconds to run the CD Play test. The default is 30 seconds.

TABLE 3-4 `audiotest` Command-Line Syntax *(Continued)*

Argument	Description
W	Shows warning messages during the Loopback test.
MF=*filename*	Selects an optional music file.
TF=*filename*	Specifies an optional tolerance file. Note: The tolerance file is used by manufacturing centers and special test facilities. Do not use this option unless you are familiar with the tolerance file format.

CHAPTER 4

Blade Support Chip Test (bsctest)

Caution – The bsctest exercises the Blade Support Chip and supporting hardware used in Sun Fire™ B100 blade systems. This includes the OpenBoot™ PROM and Time of Day (ToD) PROM chips.

Caution – If the LED subtest is selected, LEDs on the blade will change. The LEDs will return to their correct state when the test is completed.

bsctest Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

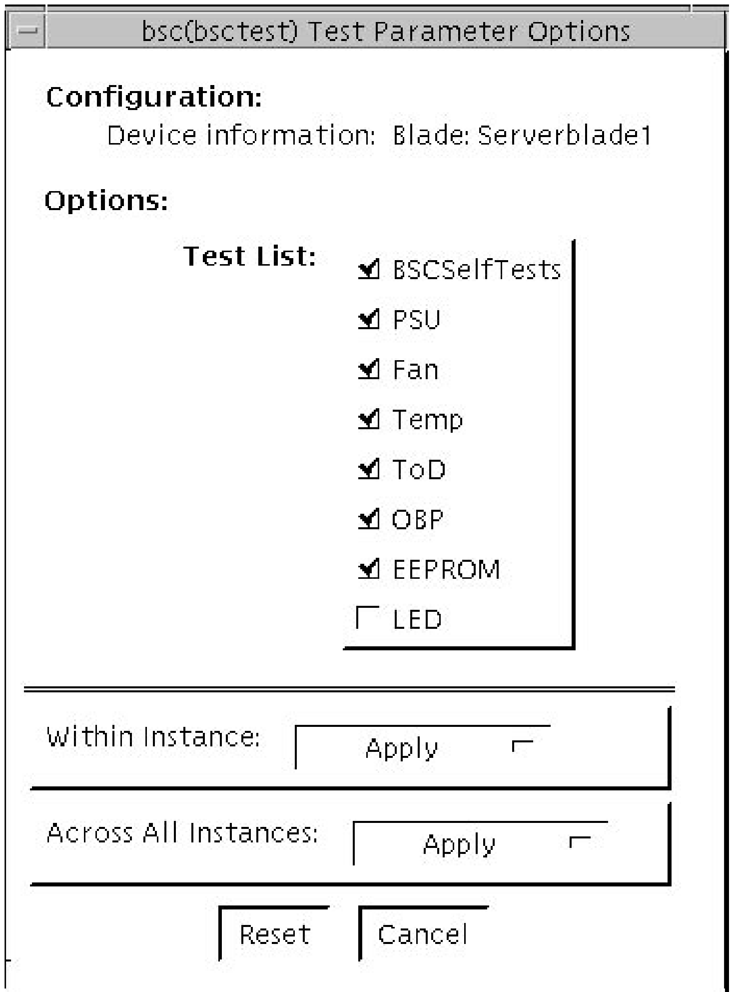


FIGURE 4-1 `bsctest` Test Parameter Options Dialog Box

TABLE 4-1 `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 4-1 bsctest Options *(Continued)*

bsctest **Options**	**Description**
OBP	Performs read-only checks of platform specific OpenBoot 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 4-2 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 4-3 bsctest Command-Line Syntax

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

CHAPTER 5

Optical Disk Drive Test (cddvdtest)

cddvdtest Description

cddvdtest verifies the functionality of optical disk drives.The disktest probe detects the media type in the drive and shows the test options for the media found. If inserted media is not supported by the drive, the disktest probe shows an error and registers options for CD-ROM as default.

Note – cddvdtest is a newly consolidated test which is used to test the optical media drives such as CD-ROM, DVD-ROM, DVD-RAM, and CD-DVD-RW drives. cddvdtest supports the same set of options for both SPARC and x86.

Volume Management

cddvdtest tests the optical drive(s) even if the volume manager is not running. If the volume manager is running and no media is installed in the drive(s), SunVTS prompts you to install media in the drive before selecting the test. The test fails if you try to run it without media in the drive.

Note – When testing rewritable media, the media can be blank or can contain the SunVTS test data. When testing write-once media, the media (such as CD-R) must be blank at the start the write test. Such media could still run multiple passes of the test, because after the first write test, the subsequent tests treat the media as read only and perform the test accordingly.

For CD-ROM and DVD-ROM drives, the test checks the unit by reading either the CD or DVD. For CD-ROMs, each track is classified as follows:

- Mode 1 uses error detection/correction code (288 bytes).
- Mode 2 uses that space for auxiliary data or as an audio track.

For rewritable CD media, the test can write one or more tracks in one test pass. `cddvdtest` writes tracks on next available space on the media. If the media is full, `cddvdtest` automatically erases the whole media and starts the next test pass from the beginning of the media.

For rewritable DVD media, the test writes only one track in one test pass (because there is only one track in DVD format). `cddvdtest` blanks the media when starting the test, if the media is not already blank.

For rewritable media types, the test verifies write, read, and other supporting functions of CD and DVD RW drives. The supported media include the following:

- CD-R (must be blank)
- CD-RW (can either be blank or contain the SunVTS test data)
- DVD-R (must be blank)
- DVD+R (must be blank)
- DVD-RW
- DVD+RW
- DVD-RAM

Note – `cddvdtest` is not a scalable test.

`cddvdtest` Hardware and Software Requirements

Requirements for `cddvdtest` differ based on the media type.

CD-ROM and DVD-ROM

The drive must have the appropriate CD-ROM or the DVD-ROM media before performing the test.

When a CD-ROM is loaded in the drive, `cddvdtest` uses CD-ROM specific options to test the drive. When a DVD-ROM is loaded, the test uses DVD-ROM specific options. Whenever you change the media in the drive, you must perform a reprobe (refer to the *SunVTS User's Guide* for details) so that the SunVTS kernel associates the correct test options with the media.

CD-RW and DVD-RW

When testing rewritable media, reprobing is required if the media is changed. To prevent accidentally erasing useful data on a media used for testing, `cddvdtest` accepts the test media only if it is blank or it contains SunVTS test data (data that is written by `cddvdtest` itself). The test checks these conditions at probe time and at the start of the test.

In case of nonblank media or media with nonSunVTS test data, the media must be blanked first by using the `cdrw` utility. Because -R or +R media can be written only once, only blank media should be used for write testing. Such media could still run multiple passes of the test because after the first write test, the subsequent tests treat the media as read only and perform the test accordingly.

To prevent media corruption, the test posts a Warning message if it stops druing the write, finalize, format, or erase phases and continues until the current operation is completed. Wait until the test completes before doing any operation on the drive.

Do not stop the test in the middle of a writing operation. Doing so may cause damage to the media in some cases. Set a limited number of passes for `cddvdtest`, set Max Passes=0 (unlimited) and stop the test manually. If a media is damaged, blank the media with `cdrw` command.

Note – DVD+RW media cannot be blanked.

The default delay between two passes for the read write media is three minutes. This setting enables the test to preserve the media by running fewer passes during long test runs. This setting also allows for stopping the test between passes.

`cddvdtest` Subtests

`cddvdtest` has different subtests for each media type.

CD-RW and DVD-RW

TABLE 5-1 `cddvdtest` Subtests for CD-RW and DVD-RW Media

Subtest	Description
Blank	Erases data on DVD-RW media and CD-RW media if not blank. This subtest does not apply to DVD+RW media because this media type can not be blanked.
Simulation	Performs a write track with laser turned off. This mode tests writing function without data written to the media.
Write	Writes to media with predefined data patterns of 0..ff hex. For CD-RW media, track can be specified as data or audio track.
Read	Reads the written data.
Compare	Compares write/read data, reports failure if miscompared.
Read Disk/Track	Reads and shows Disk Table of Contents (TOC).
Eject	Ejects media.

`cddvdtest` Options

To reach the following dialog box, 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.

`cddvdtest` has different test options for each media type.

CD-ROM Test Options

This section describes the cddvdtest options for CD-ROMs.

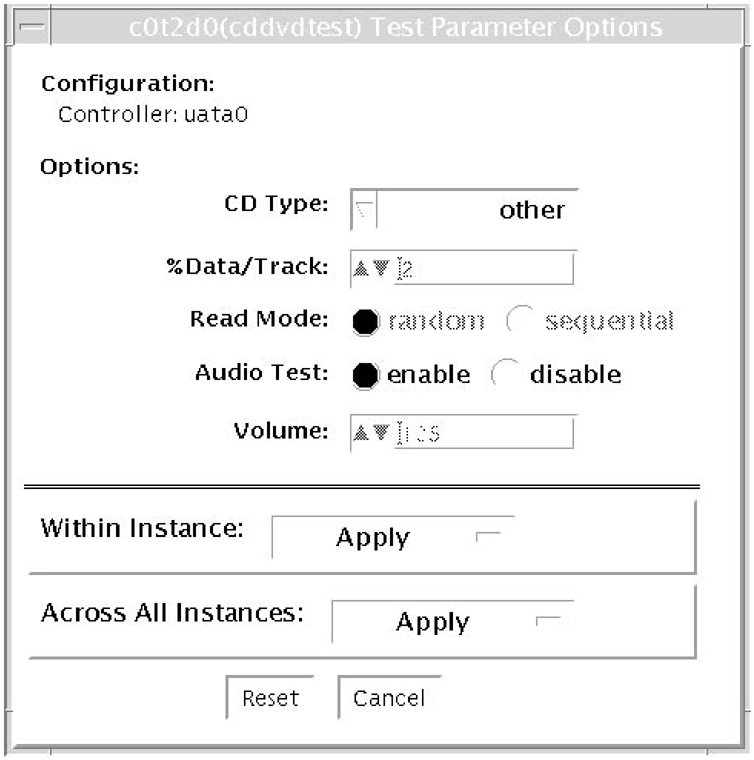


FIGURE 5-1 cddvdtest Test Parameter Options Dialog Box for CD-ROM

TABLE 5-2 `cddvdtest` Options for CD-ROM

Option	Description
CD Type	The CD Type menu lists the types of compact discs that can be tested. The choices are: Pdo, Multi-session, or Other the default CD type is other). In Connection test mode, this option has a default value of other. Note - Your choice must correspond with the disc used for testing.
% Data/Track	Tests a percentage of data on each track. Type a value between 0 and 100 in this field to indicate the percentage. In the online and connection tests this option has a default value of 2%.
Read Mode	`cddvdtest` reads the CD either in Random or Sequential mode. In Random mode, data blocks are read from random track positions; in Sequential mode, data blocks are read in sequence. For both modes, the total number of blocks read is determined by the %_of_data option. In the online and Connection tests this option has a default value of random.
Audio Test	Enables or disables the audio test. You must connect headphones or a speaker to the audio jack on the CD player to hear audio output. In the Connection test, this option has a default value of disable.
Volume	Adjusts the volume. Type a value between 0 and 255 in this field. In the online and connection tests this option has a default value of 125.

DVD-ROM Test Options

This section describes the cddvdtest options for DVD-ROMs.

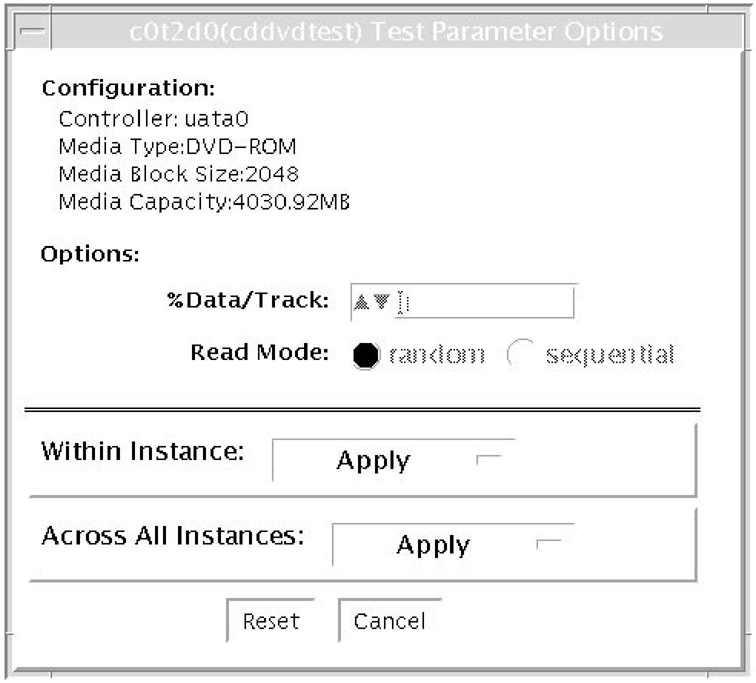


FIGURE 5-2 `cddvdtest` Test Parameter Dialog Box for DVD-ROM

TABLE 5-3 `cddvdrwtest` options for DVD-ROM

Option	Description
% Data/Track	Specifies a percentage of data to test for each track. Use a number between 0 and 100.
Read Mode	Specifies either Random or Sequential mode. Random mode reads data blocks from random track positions. Sequential mode reads data blocks in sequence. For both modes, the total number of blocks read is determined by the % Data/Track value.

CD-RW Test Options

This section describes the test options for CD-RWs.

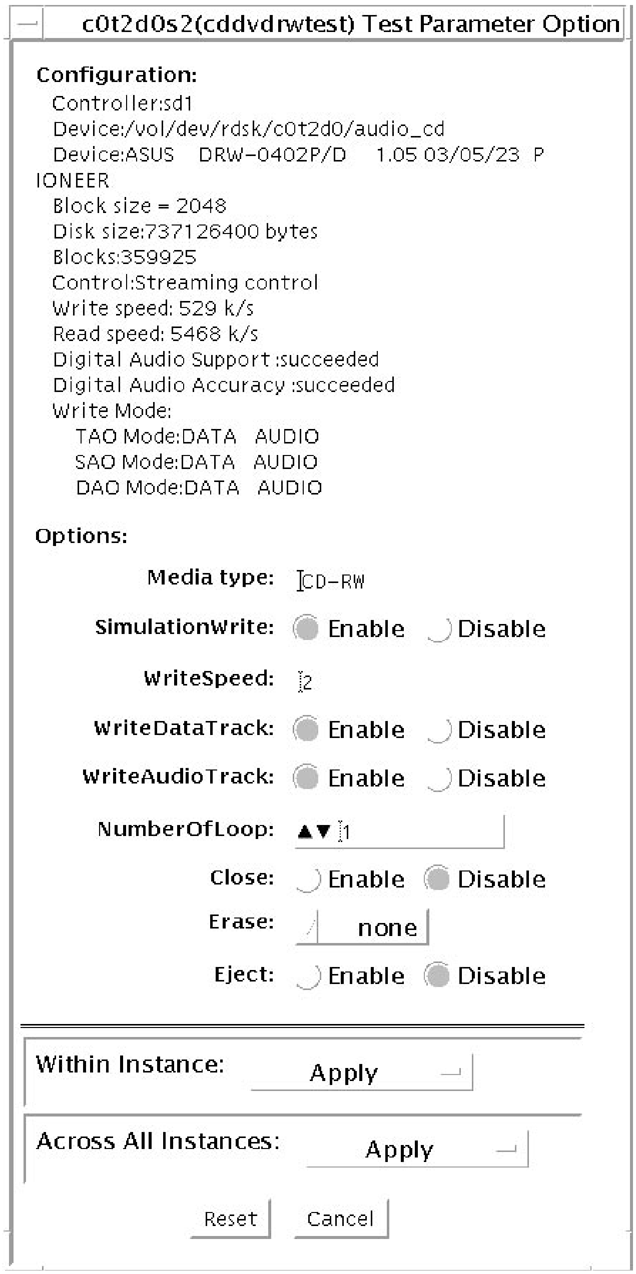


FIGURE 5-3 cddvdtest Test Parameter Options Dialog Box for CD-RW

TABLE 5-4 `cddvdtest` Options for CD-RW

Option	Description
SimulationWrite	Disables or enables simulation write.
WriteSpeed	Speed entered in terms of *nX*. Speed is set to closest approximation of *nX* allowed by the device.
WriteDataTrack	Writes a data track in one test loop.
WriteAudioTrack	Writes an audio track in one test loop.
NumberOfLoop	Number of loops in one test pass comprising data and audio track if enabled.
Close	Closes after writing. Cannot write more tracks on the media.
Erase None	Does not erase. The All option erases the whole disk.
Eject	Ejects the disk when the test is completed. Do not enable this option in a multiple pass test session.

DVD-RW Test Options

This section describes the `cddvdtest` options for DVD-RWs.

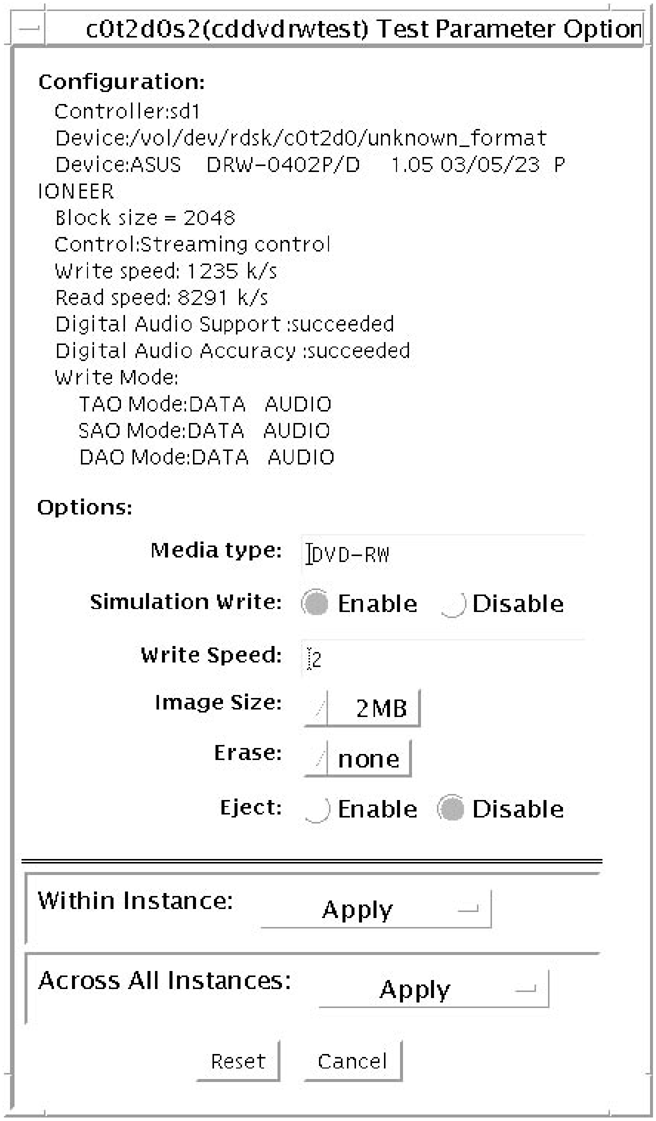


FIGURE 5-4 `cddvdtest` Test Parameter Options Dialog Box for DVD-RW

TABLE 5-5 `cddvdtest` Options for DVD-RW

Options	Description
SimulationWrite	Disables or enables simulation write.
WriteSpeed	Speed entered in terms of *nX*. Speed is set to closest approximation of *nX* allowed by the device.
ImageSize	Specifies track size, how much data is written: 2 MB, 10 MB, 2 GB, 4 GB.
Erase None	Do not erase the media.
Fast	Erases with Fast option. Only the Table of Contents is erased.
All	Erases with All option. Erases whole disk surface.
Eject	Ejects the disk when the test is completed. Do not enable this option in a multiple pass test session.

DVD-RAM Test Options

This section describes the `cddvdtest` options for DVD-RAM.

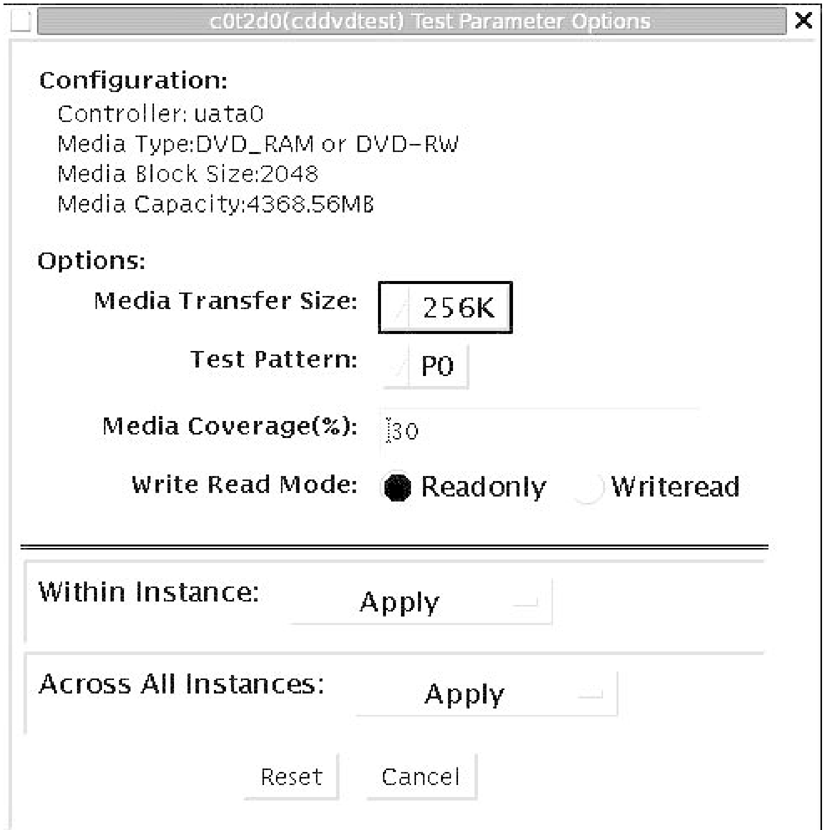


FIGURE 5-5 `cddvdtest` Test Parameter Options Dialog Box for DVD-RAM

TABLE 5-6 `cddvdtest` Options for DVD-RAM

Options	Description
Media Transfer Size	Specifies the media test size in Kbytes.
Test Pattern	Specifies the test pattern: P0, P1, or P2 (Default P0). The following options are supported: • P0 = 0000000000000000 • P1 = ff00ff00ff00ff00 • p2 = ffffffffffffffff
Media Coverage	Specifies the percentage of media coverage.
Write Read Mode	Specifies either Readonly or Writeread.

`cddvdtest` Test Modes

This test supports Connection and Functional test modes for all media types. The test behavior in each mode for the different media types is explained in the following tables.

CD-ROM Test Modes

TABLE 5-7 describes the test modes for CD-ROMs.

TABLE 5-7 Supported Test Modes for CD-ROM

Test Mode	Description
Connection	`cddvdtest` verifies that a CD-ROM drive is connected to and configured in the system.
Functional	The test registers a failure if the device is found to be busy. This is because SunVTS tests make the assumption that all the resources will be available for testing in the Functional test and the unavailability of a device is interpreted as an indication of a fault condition.

DVD-ROM Test Modes

TABLE 5-8 describes the test modes for DVD-ROMs.

TABLE 5-8 Supported Test Modes for DVD-ROM

Test Mode	Description
Connection	Requests and displays information from the drive and reads two blocks of data from the media to confirm connectivity. An error is reported if no media is loaded in the drive.
Functional	Requests and displays information from the drive, then reads data from the media based on the options that are set in the Test Parameter Options dialog box. An error is reported if no media is loaded in the drive.

CD-RW and DVD-RW Test Modes

TABLE 5-9 describes the test modes for DVD-RWs.

TABLE 5-9 Supported Test Modes for CD-RW and DVD-RW

Test Mode	Description
Connection	Shows basic drive information and supporting modes. Shows disk Table of Contents (TOC).
Functional	The following subtests are done in sequence. Erase - Erases data if DVD-RW media is not blank, or CD-RW media if full. • Simulation (optional) - Performs a write track with Laser turned off. This mode tests writing function without data written to the media. • Write - Writes to media with predefined data patterns of 0..ff hex. For CD media track can be specified as data or audio track. • Read - Reads the written data. • Compare - Compares write/read data, reports failure if miscompared. • Blank media (optional). • Read Disk/Track - Reads and shows disk table of contents. • Ejects (optional).

DVD-RAM Test Modes

TABLE 5-10 describes the test modes for DVD-RAM.

TABLE 5-10 Supported Test Modes for CD-RW and DVD-RAM

Test Mode	Description
Connection	Requests and displays information from the drive and reads two blocks of data from the media to confirm connectivity. An error is reported if no media is loaded in the drive.
Functional	Requests and displays information from the drive, then reads data from the media based on the options that are set in the Test Parameter Options dialog box. An error is reported if no media is loaded in the drive.

`cddvdtest` Command-Line Syntax

`cddvdtest` has different command-line syntax for each media type.

CD-ROM Command-Line Syntax

`/opt/SUNWvts/bin/cddvdtest` *standard_arguments* **`-o dev`**=*raw_device_name*, `mode=`*mode*, `read=random|sequential`, `data=`*%_of_data*, `vol=`*volume*, `audio=enable|disable`, `type=`*CD_type*

TABLE 5-11 CD-ROM Command-Line Syntax

Argument	Description
`dev=`*raw-device-name*	Specifies the name of the raw device to be tested.
`read=`*random* | *sequential*	Indicates random or sequential read access.
`data=`*%-of-data*	Sets the percentage of data to be tested. You can specify 0 to 100 percent.

TABLE 5-11 CD-ROM Command-Line Syntax *(Continued)*

Argument	Description
vol=*volume*	Controls the audio volume. You can specify 0 through 255. The default is 255.
audio=*enable*|*disable*	Enables or disables the audio test. You must connect headphones or a speaker to the audio jack on the CD player to hear audio output.
type=*CD-type*	Specifies the type of CD used for the test. The choices are pdo, multi-session, sunos and other. The default is other.

DVD-ROM Command-Line Syntax

/opt/SUNWvts/bin/cddvdtest *standard_arguments* **-o dev=***device_name*, **read=**random|sequential, **data=***%_of_data*

TABLE 5-12 DVD-ROM Command-Line Syntax

Argument	Description
dev=*device_name*	Specifies the name of the device to test, for example /dev/rdsk/cntndn.
read=random|sequential	Indicates random or sequential read access.
data=%_of_data	Sets the percentage of data to test. Use a number from 0 to 100.

CD-RW Command-Line Syntax

/opt/SUNWvts/bin/cddvdtest *standard_arguments* **-o dev=***cntndnsn*, **media=**CD-RW, nosim, **speed=***n*, nodata, noaudio, **loop=***n*, close, **erase=**{none, all}, eject

TABLE 5-13 CD-RW Command-Line Syntax

Argument	Description
dev=*cntndnsn*	Specifies the device under test.
media=CD-RW	Specifies the media.
nosim	Disables simulation write.
speed=*n*	Specifies the speed. Entere the speed in terms of *nX*.

TABLE 5-13 *(Continued)*CD-RW Command-Line Syntax

Argument	Description
`nodata`	Disables data track test.
`noaudio`	Disables audio track test.
`loop=`*n*	Specifies 1 to 40, the number of loops in one test pass.
`close`	Closes track after test after the test, no track can be added.
`erase=`{`none`, `all`}	`none` - Does not erase media after test complete. `all` - Erases entire disk.
`eject`	Ejects disk after test completed.

DVD-RW Command-Line Syntax

`/opt/SUNWvts/bin/cddvdtest` *standard_arguments* **`-o dev=`***cntndnsn*, **`media=`**{`DVD-RW`, `DVD+RW`}, `nosim`, **`speed=`***n*, **`imagesize=`**{`2MB`,`10MB`,`2GB`,`4GB`}, **`erase=`**{`none`, `fast`, `all`}, `eject`

TABLE 5-14 DVD-RW Command-Line Syntax

Argument	Description
`dev=`*cntndnsn*	Specifies the device under test.
`media=`{`DVD-RW`, `DVD+RW`}	Specifies the media.
`nosim`	Disables Simulation Write.
`speed=`*n*	Specifies the speed. Enter the speed in terms of *nX*.
`imagesize`=`{2MB,10MB,2GB,4GB}`	Specifies the image size used in the write/read test.
`erase=`{`none`, `fast`, `all`}	.`none` - Does not erase media after test complete. `fast` - Erases the last track added. `all` - Erases entire disk.
`eject`	Ejects the media.

DVD-RAM Command-Line Syntax

/opt/SUNWvts/bin/cddvdtest *standard_arguments* **-o dev=***cntndnsn*, rawiosize={*32K,64K,128K,256K,512K*},rawpattern={*P0,P1,P2*},**rawcover**={*1% – 100%*},**rawrw**={Readonly,Writeread}

TABLE 5-15 DVD-RAM Command-Line Syntax

Argument	Description
dev=*cntndnsn*	Specifies the device under test.
rawiosize = *32K,64K,128K,256K,512K*	Specifies the iosize for write and read operation. Default is 256K.
rawpattern = *P0,P1,P2*	The following options are supported. Default is P0. P0 = 0000000000000000 P1 = ff00ff00ff00ff00 p2 = ffffffffffffffff
rawcover = *1-100*	Specifies the media coverage in percent. Default is 30.
rawrw = Readonly, Writeread	Selects the Read and Write operations. Default is Readonly.

CHAPTER 6

Chip Multi-Threading Test (`cmttest`)

`cmttest` Description

The `cmttest` verifies the proper functioning of the multiprocessor hardware with multiple cores in one CPU. `cmttest` tests the path between the cores on the same CPU in addition to performing CPU specific testing. `cmttest` uses the Cache Coherence, Shared Memory, and RAM subtests. The Cache Coherence subtest tests the coherence among all of the Cores in a CMT (Chip Multiprocessor). The Shared Memory subtest is used to test the shared memory among all the cores in a CMT. The RAM subtest is used to test the memory. The RAM subtest covers TLB, MMU, and bus balancing. The Interrupt subtest covers the intra-core and inter-core interrupt generation and receiving logic.

Only one `cmttest` is registered. `cmttest` is present under the logical name Processor(s). There is no physical name provided. The probe routine of `cmttest` probes all CMTs in which at least two cores are online.

Note – `cmttest` was named `cmptest` in previous SunVTS releases.

`cmttest` Options

To reach the following dialog box, 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.

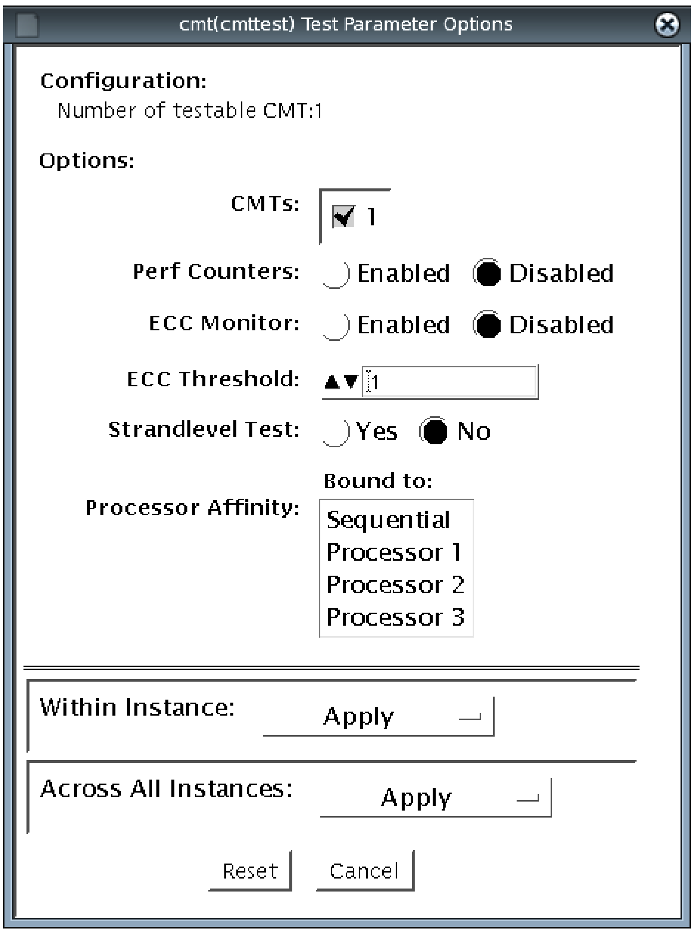


FIGURE 6-1 `cmttest` Test Parameter Options Dialog Box

The processors that can be tested are listed in the Configuration area of the menu. You can enable or disable the multiprocessing test for individual processors on this menu.

The options listed in the following table can be run alone or concurrently with other options.

TABLE 6-1 `cmttest` Options

`cmttest` **Options**	**Description**
CMTS	You can test specific CMTs by clicking Select on the check boxes to enable or disable each CMT. A check mark indicates the CMT is enabled for testing. The default setting is all CMTs enabled.
Perf Counters	By default performance monitoring is disabled. When performance monitoring is Enabled tests print memory bandwidth achieved while testing. Right now only ram subtest has the counters built in. Bandwidth calculations assume that all banks corresponding to all cpus are present and had same number of reads and writes. (Note: Perfcounter monitoring can be done on SUNW,UltraSPARC-IV processors, If user tries to enable perfCounter, and perfcounters are not supported ,on cpus the appropriate warning message is displayed, with disabling the perfcounter.)
ECC Monitor	This option is used to Enable or Disable ECC error monitoring. The default option is Disabled.
ECC Threshold	Range is [0-255].This determines how many correctable ECC errors occurred in the elapsed time before cmttest reports a test failure. The default threshold value is 1.
Strand Level Test	For processors like Niagara with multiple strands per core, the test selects one strand/core to perform testing. This behavior can be changed by setting this option to Yes. The default setting No does the most optimal testing and hence should be used in general.The interrupt subtest will not be run if this option is set to Yes. This option doesnt have any effect on non-multistranded processors such as the UltraSPARC IV+.

`cmttest` Test Modes

TABLE 6-2 `cmttest` Supported Test Modes

Test Mode	Description
Functional	Performs the full test.
Exclusive	Performs the full test.

`cmttest` Command-Line Syntax

For 32-bit configurations:

`/opt/SUNWvts/bin/cmttest` *standard_arguments*
`-o cmts=`*0+1+2...*, **`em=`***Enabled|Disabled*, **`threshold=`***[0-255]*, **`perf=`**
Enabled|Disabled, `strandleveltest=`*Yes|No*

For 64-bit configurations:

/opt/SUNWvts/bin/sparcv9/cmttest *standard_arguments* **-o cmts=***0+1+2...*, **em=***Enabled|Disabled*, **threshold=***[0-255]*, **perf=** *Enabled|Disabled*, `strandleveltest=`*Yes|No*

TABLE 6-3 `cmttest` Command-Line Syntax

Arguments	Description
cmts=*0+1+2...*	*0, 1, 2,...* mentions the CPU ID of any one core of the CMTs to be tested. To display on the GUI, CPU ID of core 0 is taken as the identifier for a CMT. For displaying the Error/INFO/LOG messages, the CPU ID of the core 0 is used. For multistranded processors such as the Niagara, if more than one strand from a core is selected, only one of them is used for testing. To override this behavior,the option `strandleveltest` must be set to Yes.
em=*Enabled|Disabled*	Enable or Disable ECC error monitoring. The default value is Disabled.
threshold=*[0-255]*	Determines how many correctable ECC errors can occur in the elapsed time before `cmttest` reports a test failure. The range is [0-255]. The default value is 1.
perf=*Enabled|Disabled*	By default performance monitoring is *Disabled*. When performance monitoring is *Enabled* tests print memory bandwidth achieved while testing. Only the RAM subtest has the counters built in. Bandwidth calculations assume that all banks corresponding to all CPUs are present and have the same number of reads and writes. Note: Perfcounter monitoring can be done on SUNW, UltraSPARC IV processors. If you try to enable perfCounter, and the perfcounters are not supported on the CPUs, the appropriate warning message is displayed and the perfcounter is disabled.
strandleveltest=*Yes|No*	This option is used for strand level testing for multi-stranded processors such as the Niagara. Selecting Yes for this option, selects all strands for testing.

CHAPTER 7

CPU Power Management Test (cpupmtest)

cpupmtest Description

cpupmtest cycles a CPU through its power management™ states. The CPU is run for user-defined periods of time from full speed, to half speed, to lowest speed, back up to half speed, and to full speed, in that order, at various levels. cpupmtest test verifies that the CPU speed changes correctly for each state.

You can run this test concurrently with device tests, to monitor whether CPU speed changes are affecting device performance. Use cpupmtest to check that all devices function correctly during different power management modes.

Note – The number of speed levels available for testing depends on the type of CPU being tested. Enter an appropriate number of arguments for your CPU: speed1, speed2 .. speedn, where speed1 is the CPU's lowest speed and speed*n* is the CPU's highest speed.

---

cpupmtest is supported on Sun Blade™ 100 and Sun Blade 1000 systems.

Note – SunVTS does not support processor sets. If processor sets are defined, you must delete the processor sets before running SunVTS.

`cpupmtest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

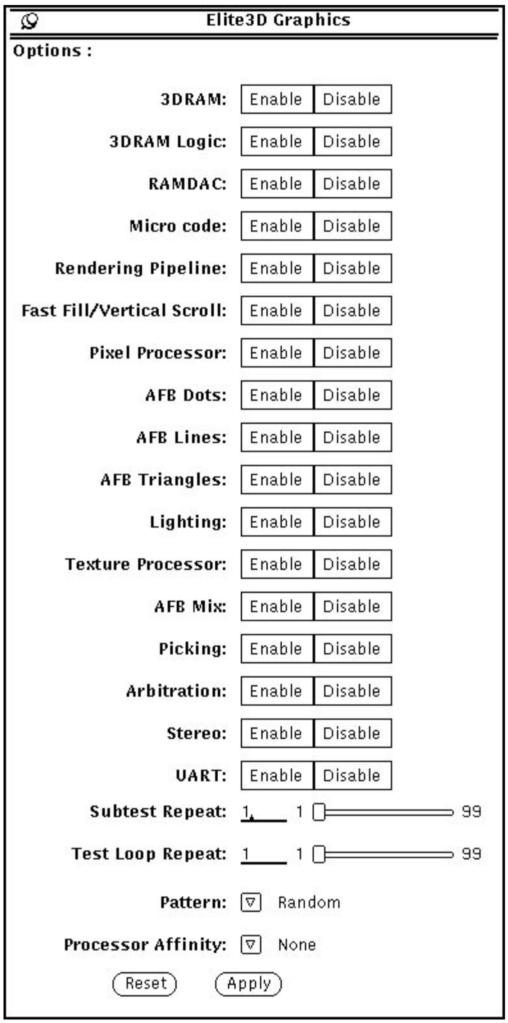


FIGURE 7-1 `cpupmtest` Test Parameter Options Dialog Box

Note – Your system might display a different number of levels in the dialog box, as the number of available levels depends on the type of CPU being tested.

TABLE 7-1 `cpupmtest` Options

Option	Description
Level1	Sets the number of minutes to test the system at its lowest speed.
Level2	Sets the number of minutes to test the system at a level higher than the lowest speed.
Level3	Sets the number of minutes to test the system at a level higher than Level2 speed.
Level *n*-1	Sets the number of mnutes to test the system at a level lower than the normal speed.
Level *n*	Sets the number of mnutes to test the system at its normal speed.
Log Power States	Records and timestamps every CPU power level change to the VTS log file (`/var/opt/SUNWvts/logs/sunvts.info`).

cpupmtest Test Modes

TABLE 7-2 cpupmtest Supported Test Modes

Test Mode	Description
Exclusive	Runs the full test.

cpupmtest Command-Line Syntax

/opt/SUNWvts/bin/cpupmtest *standard_arguments*
-o dev=*cpupm***,speed1=***mn***,speed2=***mn***,speed***n***=***mn*

TABLE 7-3 cpupmtest Command-Line Syntax

Argument	Description
dev=*cpupm*	Specifies the name of the device.
speed1=*mn*	Sets the number of minutes to test the system at its lowest speed.
speed2=*mn*	Sets the number of minutes to test the system at the next level of speed.
speed*n***=***mn*	Sets the number of minutes to test the system at its highest speed.

CHAPTER 8

CPU Test (cputest)

cputest Description

The cputest checks specific aspects of SPARC V9 processor datapath functionality.

The cputest comprises two subtests:

- g0 subtest – tests a processor's g0 register functionality. The g0 subtest is only supported on UltraSPARC-based systems.
- CUC subtest – Tests a processor's ability to correctly execute a Compress/Uncompress/Compare (cmp) command sequence on machines with the SPARC V9 architecture.
  As the CUC subtest runs, it creates four files in the /tmp/sunvts directory. The CUC subtest uses the following file naming conventions where *nnn* represents the processor unit number and *x* represents a random character string appended to the file name:
  - P*nnn*P*x* – original pattern file (size determined by the cputest File Size option)
  - P*nnn*Z*x* – Compressed version of the pattern file
  - P*nnn*U*x* – Uncompressed data from the P*nnn*Z*x* file
  - P*nnn*C*x* – Comparison data between the P*nnn*P*x* and P*nnn*U*x* files

  With the cputest file retention mode, you can control whether these files are deleted or not, so that in the event of a miscompare, you can view the contents of the files to analyze the miscompared data. The exact names of the files are displayed in the SunVTS message window whenever the files are saved, such as when there is a compression miscompare or when the file retention mode is set to save. Refer to "cputest Options" on page 60 for more details.

Caution – Do not run the CUC subtest with the retention mode set to `save` for numerous passes, otherwise the files that are saved in `/tmp/sunvts` can fill the `/tmp` capacity. If `/tmp` is mounted to the swap area, the swap space may become filled to capacity.

Note – Only one instance of `cputest` per processor is possible.

Note – When `cputest` is run with other tests, it may give the error message `exec'd program compress failed with code 1`. This can be an indication of a failed compression program, not necessarily a failed CPU. If this occurs, stop all other tests and run `cputest` alone. If the message occurs again, the CPU is failing.

Note – The `cputest` only runs on SPARC V9 systems.

Note – SunVTS does not support processor sets. If processor sets are defined, you must first delete the processor sets before running SunVTS.

`cputest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

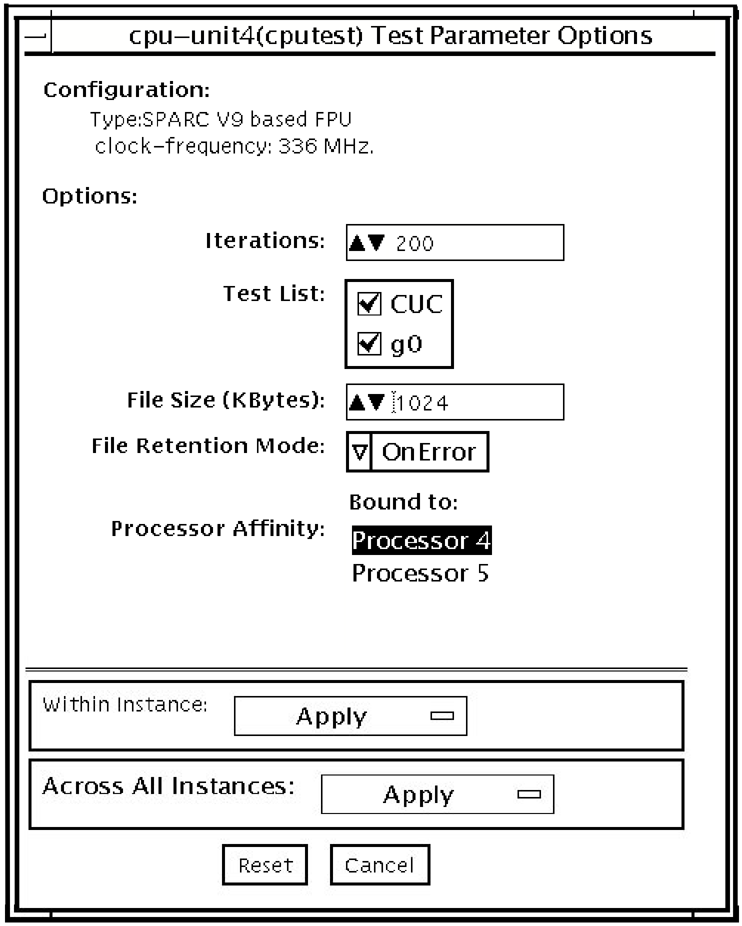


FIGURE 8-1 `cputest` Test Parameter Options Dialog Box

TABLE 8-1 `cputest` Option Dialog Box Descriptions

Option	Description
Iterations	Specifies the number of times to loop on the selected subtests. Use the up and down arrow keys to select a value from 1 to 8192. The default varies depending on the SunVTS test mode.
Test List	Specifies which subtests to run. The choices are: • CUC – The compress/uncompress/compare subtest. • g0 – The g0 register subtest. Refer to the general test description at the beginning of this chapter for subtest descriptions. If no subtest is selected, both subtests run.
File Size (KBytes)	Specifies the size of the CUC pattern file in KBytes. Select a value from 1 to 8192. The default varies depending on the SunVTS test mode.
File Retention Mode	Specifies whether the `cputest` removes the CUC pattern files or not. The choices are: • Purge – Unconditionally removes the four subtest files. • Save – Does not remove any of the four subtest files. • OnError – Removes the four subtest files unless the CUC resulted in a miscompare. In this case, do not remove the files. The default is OnError. Refer to the Caution at the beginning of this chapter regarding the Save value.
Processor Affinity	Although the Test Parameter Options dialog box displays the processor affinity Bound To selection box, the processor that corresponds to this instance of the `cputest` is determined when the SunVTS kernel probes for devices. Therefore, switching processor affinity in this dialog box is not supported.

`cputest` Test Modes

The following table describes how the `cputest` functions in the different test modes.

TABLE 8-2 `cputest` Supported Test Modes

Test Mode	Description
Connection	Both subtests are selected. The test options are fixed with the following values: • Iterations=5 • File Size=64 KBytes • File retention=OnError
Functional (Offline)	Both subtests are selectable, and all the test options are available to scale the `cputest` as needed.
Online	Supported.

`cputest` Command-Line Syntax

`/opt/SUNWvts/bin/``cputest` *standard-arguments* **`-o dev=`***device-name***`,count=`***count-number***`,test=`***testlist***`,size=`***file-size*
`,retain=`*mode*

TABLE 8-3 `cputest` Command-Line Syntax

Argument	Description
`dev=`*device-name*	Specifies the name of the device to test, for example, `cpu-unit5`
`count=`*count-number*	Defines the number of times to loop on the subtests. Use a number from `1` to `8192`. The default is `200`.

TABLE 8-3 `cputest` Command-Line Syntax *(Continued)*

Argument	Description
test=*testlist*	Specifies which subtests to run. The choices are: • CUC • g0 • CUC+g0
size=*file-size*	Specifies the size of the CUC pattern file in KBytes. Select a value from 1 to 8192. The default is 1024.
retain=*mode*	Specifies whether the cputest removes the CUC pattern files or not. The choices are: • Purge – unconditionally remove the four subtest files • Save – do not remove any of the four subtest files • OnError – remove the four subtest files unless the CUC resulted in a miscompare. In this case do not remove the files. The default is OnError. Refer to the Caution at the beginning of this chapter regarding the Save value.

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 might be available as a 32-bit test only. For more information, see "32- and 64-Bit Tests" on page 3.

CHAPTER 9

Cryptographics Test (cryptotest)

cryptotest Description

The Encryption Framework in Solaris 10 provides a user level API for access to cryptographic accelerators. This API is based on the PKCS#11 standard. Cryptographic accelerators are referred to as PKCS#11 tokens, and each cryptographic algorithm that the token accelerates is referred to as a mechanism.

cryptotest tests the mechanism supported by PKCS#11 tokens in the Solaris OS.

cryptotest supports the Sun Crypto Accelerator 500, 1000, 4000, 6000, and the UltraSPARC T1 (high-end processor with chip multithreading [CMT]) Crypto Provider. PKCS documentation is available at:
http://www.rsasecurity.com/rsalabs/PKCS

TABLE 9-1 Definitions of the Mechanisms Tested by cryptotest

Algorithm	Description
AES	CTR and CBC modes
Diffie-Hellman	Crypto mechanism
DSA	Digital signature algorithm
DES	Data encryption standard as defined in FIPS PUB 46-3

TABLE 9-1 Definitions of the Mechanisms Tested by `cryptotest` *(Continued)*

Algorithm	Description
MD5 RSA	Data Security MD5 message-digest algorithm
RSA	Public key cryptosystem
SHA1	Secure hash algorithm
RNG	Random number generator algorithm

`cryptotest` Subtests

TABLE 9-2 cryptotest Subtests

Subtest	Description
AES	Tests CTR and CBC modes
Diffie-Hellman	Tests the crypto mechanism
DES	Tests DES bulk encryption
3DES	Tests 3DES bulk encryption
RSA	Tests RSA public and private keys
DSA	Tests DSA signature verification
RNG	Tests random number generation

`cryptotest` Options

To reach the following dialog boxes, right-click on the test name in the System Map and select Test Parameter Options. Because graphics tests can test multiple types of frame buffers, the test name that is displayed will correspond to the particular framebuffer being tested. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system might not include the device appropriate to this test. Refer to the *SunVTS User's Guide*.

`mcatest` Options

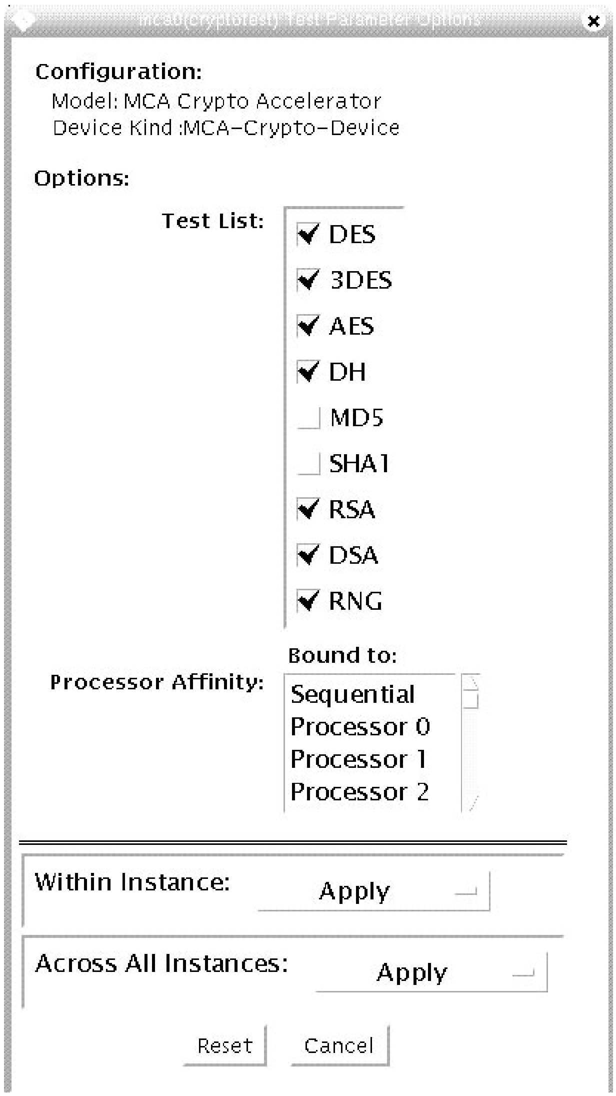


FIGURE 9-1 `mcatest` Test Parameter Options Dialog Box

TABLE 9-3 `mcatest` Options

Option	Description
AES	Tests CTR and CBC modes
Diffie-Hellman	Tests the crypto mechanism
DES	Tests DES bulk encryption
3DES	Tests 3DES bulk encryption
MD5	Data security MD5 message-digest algorithm.
SHA1	Secure hash algorithm.
RSA	Tests RSA public and private keys

vcatest Options

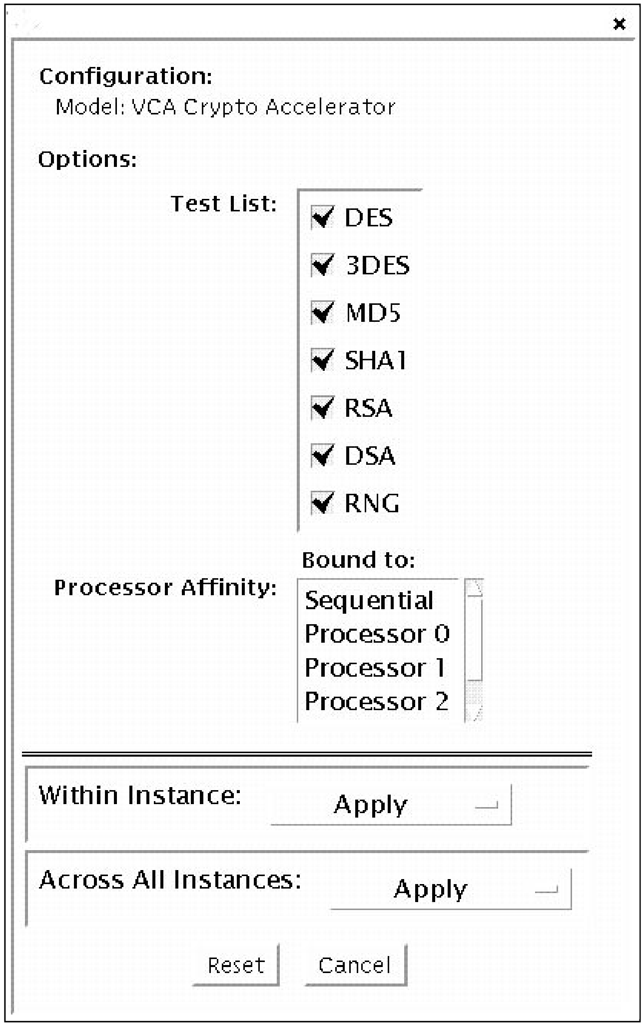


FIGURE 9-2 vcatest Test Parameter Options Dialog Box

TABLE 9-4 `vcatest` Options

Option	Description
DES	Tests DES bulk encryption
3DES	Tests 3DES bulk encryption
MD5	Data security MD5 message-digest algorithm.
SHA1	Secure hash algorithm.
RSA	Tests RSA public and private keys
DSA	Tests DSA signature verification
RNG	Tests random number generation

`dcatest` Options

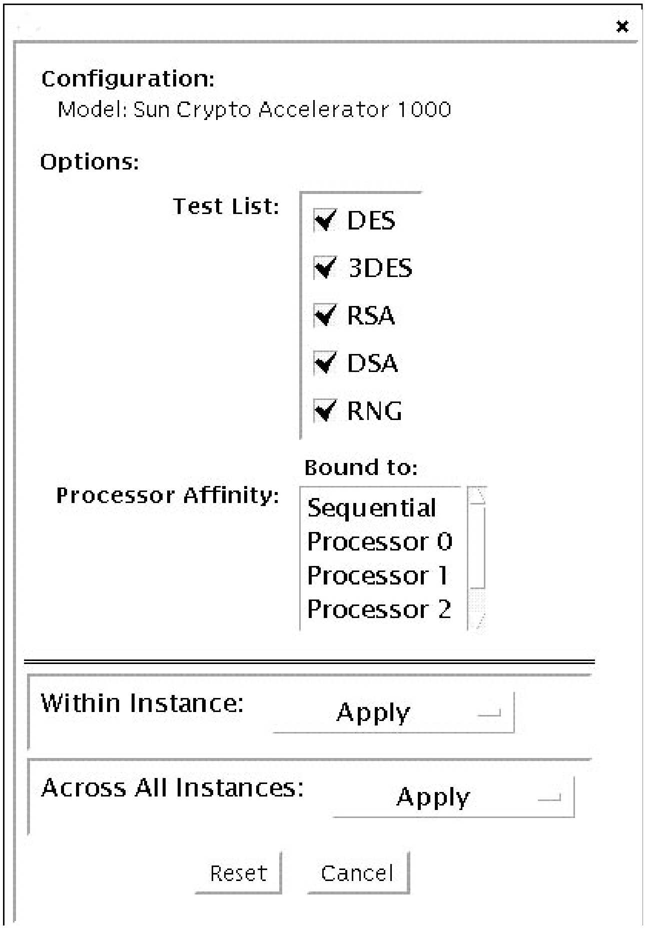


FIGURE 9-3 `dcatest` Test Parameter Options Dialog Box

TABLE 9-5 `dcatest` Options

Option	Description
DES	Tests DES bulk encryption
3DES	Tests 3DES bulk encryption
RSA	Tests RSA public and private keys
DSA	Tests DSA signature verification
RNG	Tests random number generation

cryptotest Test Modes

TABLE 9-6 cryptotest Supported Test Modes

Test Mode	Description
Functional	Runs the full set of tests.

cryptotest Command-Line Syntax for mcatest

/opt/SUNWvts/bin/sparcv9/cryptotest -f -o dev=mca2, **tl=***testlist*

TABLE 9-7 cryptotest Command Line Syntax for vcatest

Option	Description
dev=mca*N*	Specifies the instance of the device to test such as mca0 or mca2. Defaults to mca0 if not included. *N* specifies the placement of the instance number of the device being tested.
tl=*testlist*	Specifies the list of subtests to be performed. The subtests for tl are separated by the + (plus) character. The supported subtests are AES, Diffie-Hellman (DH), DES, 3DES, DSA, RSA, and RNG, so tl=AES+DH+DES+3DES+DSA+RSA+MD5+SHA1+RNG enables all subtests. You can also insert tl=all, which performs all tests. Defaults to all if no subtests are specified.

cryptotest Command-Line Syntax for vcatest

/opt/SUNWvts/bin/sparcv9/cryptotest -f -o dev=vca2, **tl=***testlist*

TABLE 9-8 `cryptotest` Command Line Syntax for `vcatest`

Option	Description
dev=vca*N*	Specifies the instance of the device to test such as vca0 or vca2. Defaults to vca0 if not included. *N* specifies the placement of the instance number of the device being tested.
tl=*testlist*	Specifies the list of subtests to be performed. The subtests for tl are separated by the + (plus) character. The supported subtests are DES, 3DES, DSA, RSA, and RNG, so tl=DES+3DES+DSA+RSA+MD5+SHA1+RNG enables all subtests. You can also insert tl=all, which performs all tests. Defaults to all if no subtests are specified.

`cryptotest` Command-Line Syntax for `dcatest`

/opt/SUNWvts/bin/sparcv9/cryptotest -f -o dev=vca2|dca2, **tl=***testlist*

TABLE 9-9 `cryptotest` Command Line Syntax for `dcatest`

Option	Description
dev=dca*N*	Specifies the instance of the device to test such as dca0 or dca2. Defaults to dca0 if not included. *N* specifies the placement of the instance number of the device being tested.
tl=*testlist*	Specifies the list of subtests to be performed. The subtests for tl are separated by the + (plus) character. The supported subtests are DES, 3DES, DSA, RSA, and RNG, so tl=DES+3DES+DSA+RSA+MD5+SHA1+RNG enables all subtests. You can also insert tl=all, which performs all tests. Defaults to all if no subtests are specified.

CHAPTER 10

Disk and Diskette Drives Test (disktest)

disktest Description

disktest verifies the functionality of hard drives and diskette drives using three subtests (see TABLE 10-1)—Media, File System, and Asynchronous I/O.

Most disk drives are supported, including SCSI disks, native or SCSI floppy disks, and IPI drives. 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. 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 specify that SunVTS premounts all mountable partitions, set the environment variable BYPASS_FS_PROBE to 0 (zero) before starting SunVTS. Premounting can be disabled by disabling 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 the partition as superuser under the name /disktest_c0t3d0s0.

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

Caution – Running the Media subtest on a disk partition in the WriteRead mode might cause data corruption if the same partition is being used by other applications. Run SunVTS in the offline mode only when there are no other applications running.

---

disktest tests the disktest drive 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 might 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.

When a large number of disktest instances are run in write/read mode, tests might fail with messages similar to the following.

```
03/22/03 03:33:40 ctech140 SunVTS5.1ps2: VTSID 8011 disktest.FATAL
c1t0d0: "Failed lock mtab semaphore. "semop" system call failure,
errmsg: Invalid argument." Probable_Cause(s): <disktest instances
exceeds system semaphore operation limitation (default system
limit for seminfo_semmnu = 30)><System software error>
Recommended_Action(s): <Add the line "set semsys:seminfo_semmnu=
0x100" to your /etc/system file and reboot the machine> <If the
problem persists, call your authorized Sun service provider.
```

To avoid this issue, add the following entry to the /etc/system file and reboot the system.

```
set semsys:seminfo_semmnu=0x100
```

disktest Subtests

The following table describes the disktest subtests:

TABLE 10-1 disktest Subtests

Subtest	Description
Media subtest	Verifies the disk media by enabling users to run disktest in different modes such as ReadOnly, ReadCompare, and WriteRead. The Media subtest treats the disk partition as one large piece of contiguous data. In 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 three modes can run two different methods of disk testings. These are Synchronous I/O and Asynchronous I/O. SyncIO – Reads and writes data using Read/Write system calls in a sequential fashion until the specified percentage of media is covered. AsyncIO – Reads and writes data using aio library calls such as aioread() and 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.

disktest Test Options

To reach the following dialog box, 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.

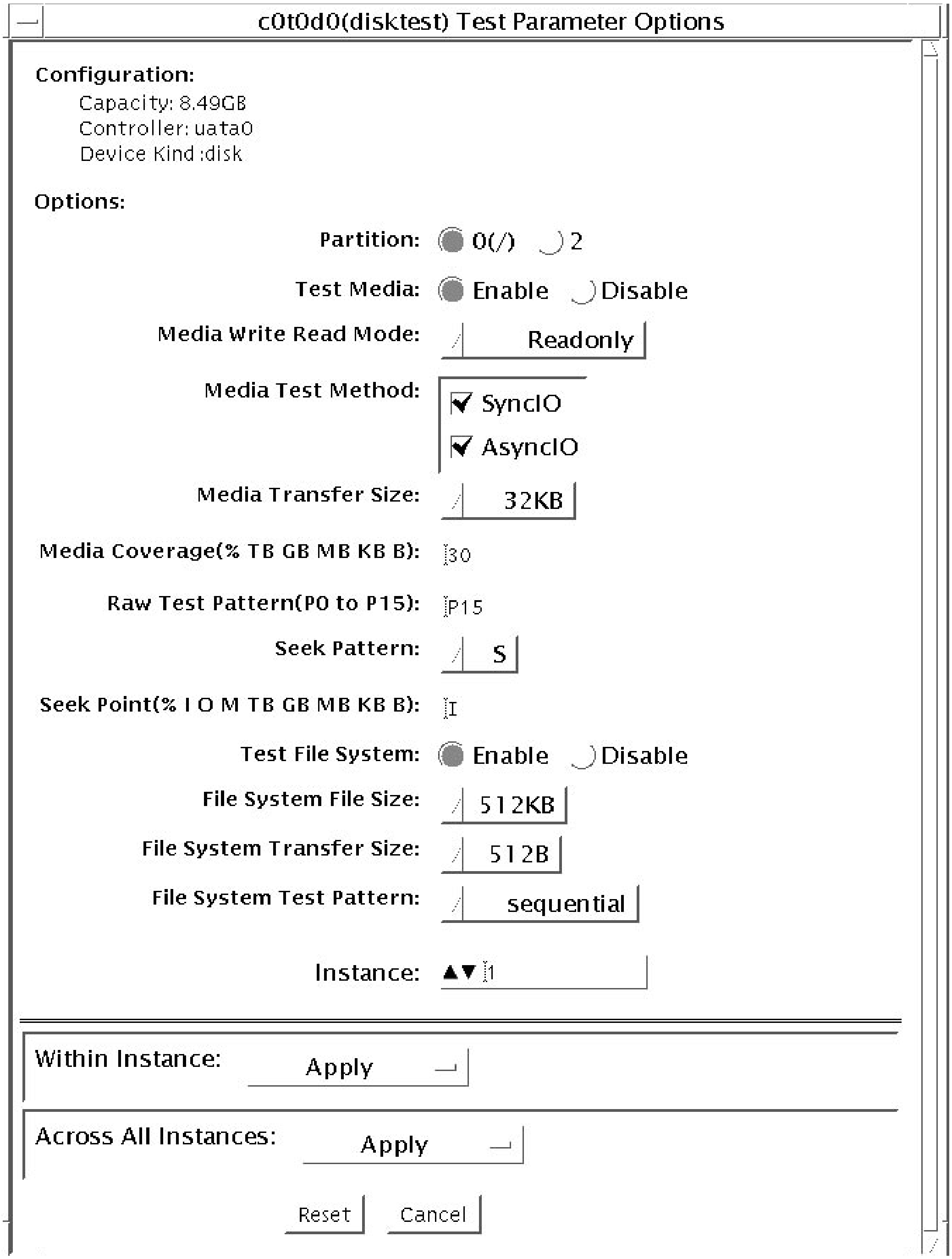


FIGURE 10-1 `disktest` Test Parameter Options Dialog Box

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

TABLE 10-2 `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	Selects Read-Only or Compare after Read, or Read after Write.
Media Test Method	Selects the media test methods (SyncIO and AsyncIO).
Media Coverage (% TB, GB, MB, KB, B)	Tests all or part of a partition (in percentage or in any of TB, GB, MB, KB, B units)
Raw Test Pattern (P0 to P16)	Specifies 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 • P16 – SATA Test Patterns
Seek Pattern	Specifies the pattern of the disk head movement. • S – Sequential • SR – Sequential Reverse • LS – Low Power Sequential • R – Random • LB – Low Power Butterfly • LR – Low Power Reverse Butterfly • AB – Actuator Butterfly • AR – Actuator Reverse Butterfly

TABLE 10-2 `disktest` Configurations and Options *(Continued)*

`disktest` **Options**	**Description**
Seek Point (%, I, O, M, TB, GB, MB, KB, B)	Specifies the seek point offset for the I/O. You can specify the offset as a percentage or any of TB, GB, MB, KB, B or and I, M, O (Initial, Middle, Outer).
Media Transfer Size	Displays the transfer size of the Media subtest.
Test File System	Selects 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.
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 – [2 KB] • Test File System – [Disable] (fixed to Disable)

TABLE 10-2 `disktest` Configurations and Options *(Continued)*

`disktest` **Options**	**Description**
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) – [10~] (fixed to 10%) • Media Transfer Size – [2KB~] (fixed to 2 KB) • Test File System – [Disable~] (fixed to Disable) • Media Test Method – [SyncIO] [AsyncIO] • Raw Test Pattern – [P15~] (fixed to P15) • Seek Pattern – [S~] (fixed to S) • Seek Point (% TB GB MB KB B) – [I~] (fixed to I)
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 P16) • 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] • Seek Pattern – [S~] (fixed to S) • Seek Point (% TB GB MB KB B) – [I~] (fixed to I)
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 P16) • 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 10-3 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 diskette test can be run in Functional test mode. In Functional mode, disktest performs an additional subtest (Write/Read device buffer subtest) for enclosures.
Online	SunVTS disktest runs the Read Only rawtest with fixed transfer size and fixed rawtest pattern. Both SyncIO and AsyncIO test methods are available. The File system subtest is disabled in the Online test mode. Only one disktest instance could be run in the Online test mode.

disktest Command-Line Syntax

/opt/SUNWvts/bin/disktest *standard-arguments* [-o partition=[*0-7*], rawsub=[*E(nable)*|*D(isable)*],rawrw=[*Readonly*|*CompareRead*|*WriteRead*], rawiosize=*number* [*KB*|*kb*|*random*],rawcover=*number* [*TB*|*GB*|*MB*|*KB*|*B*|*tb*|*gb*|*mb*|*kb*|*b*|],rawpattern=*P*|*0x<8 number pattern* [*0*|*1*|*2*|*3*|*4*|*5*|*6*|*8*|*9*|*10*|*11*|*12*|*13*|*14*|*15*],method=[*AsyncIO+SyncIO*],fssub= [*E(nable)*|*D(isable)*],fssize=*number* [*K*|*KB*|*M*|*MB*|*k*|*kb*|*m*|*mb*],fsiosize=*number* [*K*|*KB*|*B*|*k*|*kb*|*b*],fspattern= [*seq(quential)*|*0x0(0000000)*|*0xf(fffffff)*|*0xa(5a5a5a5)*|*0x5(a5a5a5a)*|*ran(dom)*|*0xd(b6db6* *db)*],dev=*device-name*, seekpattern=[*S* | *SR* | *LS* | *R* | *LB* | *LR* | *AB* | *AR*],seekpoint=*i(nner)* | *m(iddle)* | *o(uter)* | *number* [*TB* | *GB* | *MB* | *KB* | *B* | *tb* | *gb* | *mb* | *kb* | *b*],runtime=*number M*, errcnt=*number*, wrbuftest=*Enable* | *Disable*]]

TABLE 10-4 disktest Command-Line Syntax

Argument	Description
partition=*0-7*	Specifies the partition number as follows: • *n* – The partition number (slice number), usually *0-7* • *mount-point* – 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*
rawiosize= *number[...	KB	kb...]	random*	Specifies the media size to transfer. The block size can be specified in kilobytes. For example, 2K to 512K. For example: **rawiosize=**9
rawcover= *number	number[TB	GB	MB	KB	B	tb	gb	mb	kb	b]*	Specifies media coverage from 0-100 (percentage) of the partition. Media Coverage can also be specified in the following units—TB, GB, MB, KB and B. For example: **rawcover=***40* or **rawcover=***4GB*

TABLE 10-4 `disktest` Command-Line Syntax *(Continued)*

Argument	Description
rawpattern=*P(0-16)|0x8 digit data pattern*	`rawpattern` could be specified as a predefined pattern set, *P(0-16)*, 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 • P16 - SATA Test Patterns For example: **rawpattern=***P1*
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*
fssub=*E(nable)|D(isable)*	Enables or disables the File System subtest. File System subtest runs on a mounted partition with a File System.
fsiosize=*number[K|KB|B|k|kb|b]*	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`

TABLE 10-4 `disktest` Command-Line Syntax *(Continued)*

Argument	Description
fspattern= [*seq(quential)* | *0x0(0000000)* | *0xf(fffffff)* | *0 xa(5a5a5a5)* | *0x5(a5a5a5a)* | *ran(dom)* | *0xd (b6db6db)*]	Specifies the file system data pattern as sequential or random or one of the patterns selected from the list. *[seq(uential)* | *0x0(0000000)* | *0xf(fffffff)* | *0xa (5a5a5a5)* | *0x5(a5a5a5a)* | *ran(dom)* | *0xd(b6db6db)]* For example: • **fspattern=***0xa* • **fspattern=***seq*
dev=*device-name*	Specifies the name of the disk to be tested. For example: c0t3d0.
seekpattern= [*S* | *SR* | *LS* | *R* | *LB* | *LR* | *AB* | *AR*]	`disktest` supports the following pattern types: • `S` – Sequential • `SR` – Sequential Reverse • `LS` – Low Power Sequential • `R` – Random • `LB` – Low Power Butterfly • `LR` – Low Power Reverse Butterfly • `AB` – Actuator Butterfly • `AR` – Actuator Reverse Butterfly For example: `seekpattern=`*S*
seekpoint= *[i(nner)* | *m(iddle)* | *o(uter)* | *number* [*TB* | *GB* | *MB* | *KB* | *B* | *tb* | *gb* | *mb* | *kb* | *b*]	Specifies the seek-point for the I/O. This could be specified either in terms of the range - inner, middle and outer. Or in terms of absolute seek location. The absolute location is specied by a number followed by any of the following units [TB | GB | MB | KB | B | tb | gb | mb | kb | b]. For example: • `seekpoint=`*I*, starts the I/O from block 1. • `seekpoint=`*M*, starts the I/O from middle offset of the partition.
runtime=[*number M]*	Specifies the test run time in minutes.
errcnt=*number*	Specifies the maximum error count.
wrbuftest=*[Enable* | *Disable*]	Enables or disables the Write/Read Buffer test.

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 the SyncIO method. The coverage specified is 30% with 512 KB 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
```

CHAPTER 11

Netra-CT 820 DMC Test (dmctest)

dmctest Description

The dmctest tests major components in the drawer management controller (DMC) card of the Netra™ CT server. The components tested on the card include flash memory, SDRAM, FPGA, RTC, I^2C devices, UART chip, Fast Ethernet controller and PHY, Muxes, Alarm LEDs and so on. This test supports both the active (dmcatest) and standby (dmcstest) DMC card testing. The traces to be tested on the board include address bus, data bus, I2C bus, KCS, IPMI bus, interrupt lines, and the cPCI connector.

dmctest Subtests

dmctest has the following subtests:

- Ethernet test

  The Ethernet test performs tests on the SCC Ethernet device. The test covers Phy, Internal, and External loopback. The Ethernet Ping test uses the FEC port. The Ethernet test is supported on both DMC active and DMC standby cards, and performs the following tests: External loopback test, Internal loopback test, Phy test, Ping test. The Ethernet test requires an external loopback connector to perform the External loopback test.

- Flash test

  This test reads the entire 24-Mbyte flash area and performs a checksum test. The test is supported on both DMC active and standby cards.

- Alarm Port test

  This test toggles all four alarm ports, Critical, Major, Minor, and Warning, and reports Pass or Fail status. The test is supported on both DMC active and standby cards.

- Health test

  This test checks systems health and reports the state of each slot. `dmctest` reports the slot status, whether the slot is healthy or not-healthy, whether the slot is empty or occupied, whether the slot is powered-on or powered-off. The test reports their status according to the following table:

TABLE 11-1

Board Device	Health	Board Select	Status	Report
0	0	0	Slot occupied, but powered off	
0	0	1	Slot Empty	
0	1	0	Unpowered slot reporting healthy	Warning
0	1	1	Empty slot reporting healthy	Warning
1	0	0	Board select drive problem	Warning
1	0	1	Slot unhealthy	
1	1	0	Board select drive	Warning
1	1	1	Slot powered on	

- PMI test

  This test checks for the IPMI connectivity by sending IPMI commands to all slots that are occupied, then waits for their response. The IPMI command is sent to 18 node cards, both switch cards and the DMC standby card. This test performs up to four retries whenever it fails to receive a response. The retries are according to PICMG IPMI (Intelligent Platform Management Interface) Specification. The IPMI test that runs on the DMC active card performs tests on all slots that are occupied.

They are 18 node slots, two switch cards, and the DMC standby card. Whereas the IPMI test that runs on the DMC standby card tests only the IPMI connectivity between the DMC active and DMC standby cards.

- RTC test

  This test runs on both DMC cards. The RTC test checks the real time clock device and reports pass or fail status. The RTC test then displays the time of the day the status was obtained from RTC.

- Fan Status test

  This test checks the status for all three banks of three fans, and reports which fans are present and which fans are good. This test runs only on DMC active cards.

- Fan Tachometer test

  This test checks the Tachometer for all nine (three banks of three) fans. The test reports RPM value of each fan and whether a fan is operating at normal speed or at warning or critical speed. This test runs only on DMC active cards.

- FRU Presence test

  This test checks the status for all FRU devices available on Netra CT 820 systems. This test runs on DMC active cards only.

FRU Device	Number of FRUs
Midplane	2
DMC Cards	2
Fan Trays	3
Switch Cards	2
Node CP2300 Cards	18

- FRUID Checksum test

  The FRUID Checksum test traverses through all available FRU devices on Netra CT 820 systems and checks for the FRU data. The test reads FRU buffer, computes the checksum, and compares that with the checksum that is read from FRU devices. The test then reports pass or failure status. This test performs up to four retries whenever it fails to read requested FRU data. The retries loop performs according to the IPMI specification.

- I^2C Temperature test

  This I^2C Temperature option performs tests on I^2C devices. The test obtains sensor temperatures and reports the device status as Pass, Warning, or Critical.

  There are eight midplane sensors and three fan sensors. This test reports pass status when any of the midplane or fan senses temperatures are near air temperature and rise from inlet to outlet at 55C ambient. Therefore a midplane or fan sensor emits a warning if it reaches 75C (20C over 55C inlet), and it reports critical at 85C (30C over 55C maximum inlet air temperature). This test runs only on DMC active cards.

- Power Supply Status test

  This test reports the status for all eight power supplies. The status contains whether the Power Supply is powered on or off, whether the power supply is Derating or not, or whether it is Healthy or Not-Healthy.

- Power Supply On/Off test

  This test toggles all eight power supplies by turning them off and on. The test toggles one power supply at a time, starting from the first power supply.

Note – Enable the Power Supply On/Off test for only a limited number of runs; this test might degrade the life cycle of power supplies. This test runs on DMC active cards only.

The single serial communication port on DMC cards is used for the console. The serial port test is covered to enable entering commands in the console window and receiving responses from CLI commands such as `showhealth` or `showipmode`, and `ping`.

`dmcatest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

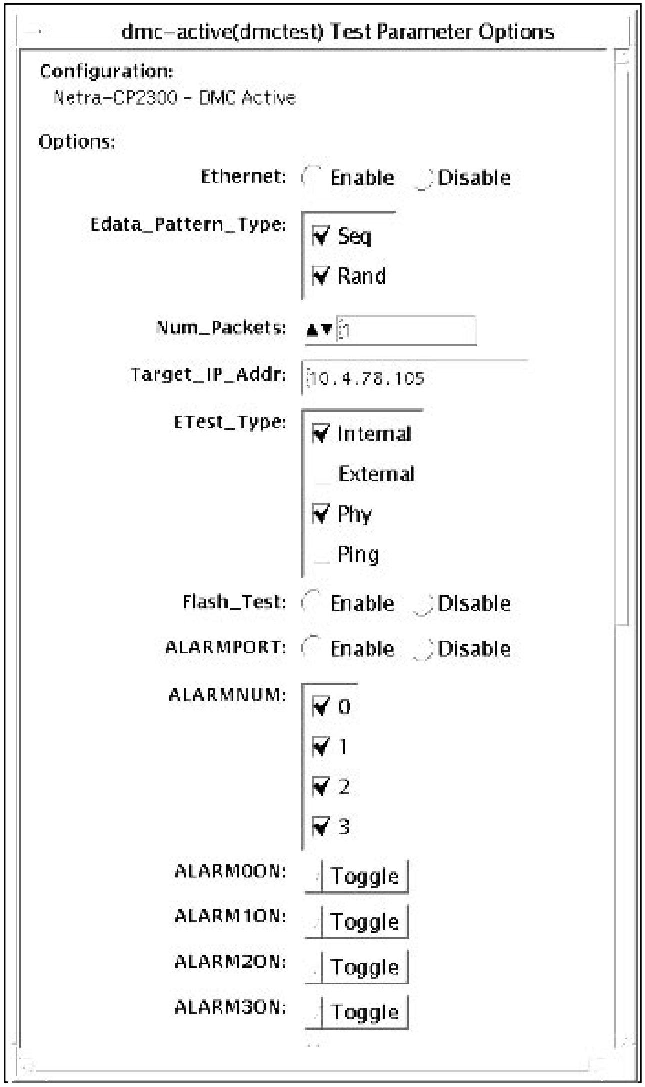


FIGURE 11-1 `dmcatest` (Active) Test Parameter Options Dialog Box (Top of form)

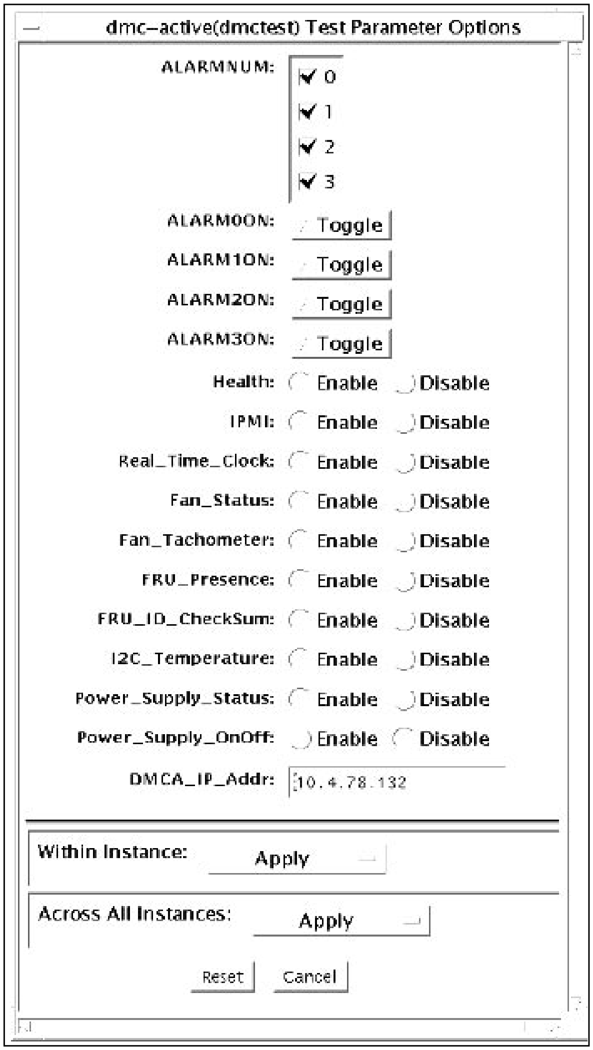


FIGURE 11-2 `dmcatest` (Active) Test Parameter Options Dialog Box (Bottom of form)

TABLE 11-2 `dmcatest` Options

`dmctest` **Options**	**Description**
Ethernet	Enables or disables the Ethernet test.
Edata Pattern Type	Indicates pattern type. Seq = Sequence, Rand = Random.
Num Packets	Indicates the number of packets to be tested.

TABLE 11-2 `dmcatest` Options *(Continued)*

`dmctest` **Options**	**Description**
Target IP Address	Required IP address for Ethernet ping test.
Etest Type	Indicates the type of data to test.
Flash Test	Enable or disable the flash test.
ALARMPORT	Enable or disable the alarmport.
ALARMNUM	Indicates the alarm number.
ALARM00N	Toggles the alarm00n option on or off.
ALARM10N	Toggles the alarm10n option on or off.
ALARM20N	Toggles the alarm20n option on or off.
ALARM30N	Toggles the alarm30n option on or off.
Health	Enable or disable the health test.
IPMI	Enables or disables the ipmi test.
Real Time Clock	Enables or disables the real time clock test.
Fan Status	Enables or disables the fan status test.
Fan Tachometer	Enables or disables the fan tachometer test.
FRU Presence	Enables or disables the FRU presence test.
FRU ID CheckSum	Enables or disables the FRU ID checksum test.
I2CTemperature	Enables or disables the i2c tempature test.
Power Supply Status	Enables or disables the power supply status test.
Power Supply On/Off	Enables or disables the power supply on/off test. The default is Disable.
DMCA IP Address	Required DMCA IP address

`dmcstest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

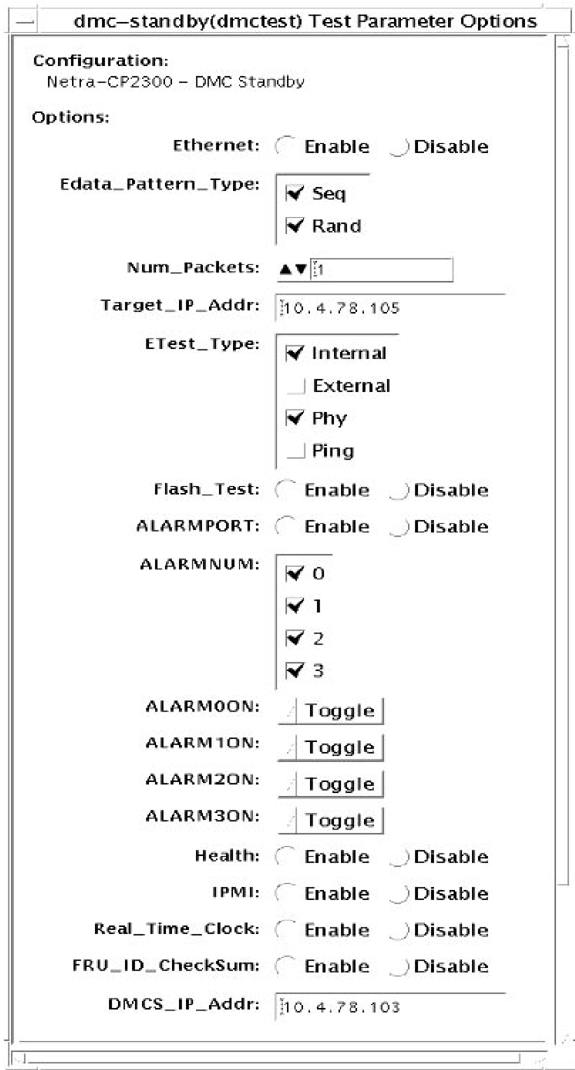


FIGURE 11-3 `dmcstest` (Standby) Test Parameter Options Dialog Box

TABLE 11-3 `dmcstest` Options

`dmctest` **Options**	**Description**
Ethernet	Enables or disables the Ethernet test.
Edata Pattern Type	Indicates the pattern type. Seq = Sequence, Rand = Random.
Num Packets	Indicates the number of packets to be tested.

TABLE 11-3 `dmcstest` Options *(Continued)*

`dmctest` **Options**	**Description**
Target IP Address	Required IP address for Ethernet ping test.
Etest Type	Indicates the type of data to test.
Flash Test	Enable or Disable the Flash test.
ALARMPORT	Enable or Disable the ALARMPORT.
ALARMNUM	Indicates the Alarm Number.
ALARM00N	Toggles the alarm00n option on or off.
ALARM10N	Toggles the alarm10n option on or off.
ALARM20N	Toggles the alarm20n option on or off.
ALARM30N	Toggles the alarm30n option on or off.
Health	Enables or disables the Health test.
IPMI	Enables or disables the IPMI test.
Real Time Clock	Enables or disables the Real Time Clock test.
FRU ID CheckSum	Enables or disables the FRU ID Checksum test.
DMCS IP Address	Required DMCS IP address.

`dmctest` Test Modes

The following test modes are supported by `dmcatest` and `dmcstest`.

TABLE 11-4 `dmctest` Supported Test Modes

Test Mode	**Description**
Function	Establishes communication with the DMC Active Diag Daemon Actor, initiates all subtests that are enabled, and reports pass or fail status.
Connection	Attempts to establish communication with DMC Active Diag Daemon Actor and reports pass or fail status.

`dmcatest` Command-Line Syntax

/opt/SUNWvtshm/bin/dmctest -o option=*value*, **enet=**{*Enable*| *Disable*}, **epattype=**{*Seq+Rand*}, **epkts=**{*NUMERIC*|*1,5*}, **target=**{*IP_Address*}, **etest=**{*Internal+External+Ping+Phy*}, **flash=**{*Enable* | *Disable*}**, aport=**{*Enable* | *Disable*}, **anum=**{*0+1+2+3*}, **a0on**={*On*|*Off*|*Toggle*}, **a1on**={*On*|*Off*|*Toggle*}, **a2on**={*On*|*Off*|*Toggle*}, **a3on**={*On*|*Off*|*Toggle*}, **health**=*{Enable* | *Disable}*, **ipmi**={*Enable* | *Disable*}, **rtc**={*Enable* | *Disable*}, **fanstatus**={*Enable* | *Disable*}, **fantac**={*Enable* | *Disable*}, **frupres**={*Enable* | *Disable*}, **fruidchksum**={*Enable* | *Disable*}, **i2ctemp**={*Enable* | *Disable*}, **psupplystatus**={*Enable* | *Disable*}, **powersupply**={*Enable* | *Disable*}, **dmca_ip**=*IP_Address*, **dmcs_ip**=*IP_Address,* **dev**={*dmc-active* | *dmc-standby*}

TABLE 11-5 `dmcatest` Command-Line Syntax

Argument	Description
enet={*Enable*| *Disable*}	Enables or disables the Ethernet test.
epattype={*Seq+Rand*}	Indicates pattern type. Seq = Sequence, Rand = Random.
epkts={*NUMERIC*|*1,5*}	Indicates the number of packets to be tested.
target={*IP_Address*}	Required IP address for the Ethernet Ping test.
etest={*Internal+External+Ping+Phy*}	Indicates the type of data to test.
flash={*Enable* | *Disable*}	Enable or Disable the Flash test.
aport={*Enable* | *Disable*}	Enable or Disable the ALARM port.
anum={*0+1+2+3*}	Indicates the Alarm Number.
a0on={*On*|*Off*|*Toggle*}	Toggles the `a0on` option on or off.
health=*{Enable* | *Disable}*	Enables or disables the Health test.
ipmi={*Enable* | *Disable*}	Enables or disables the IPMI test.
rtc={*Enable* | *Disable*}	Enables or disables the Real Time Clock test.
fanstatus={*Enable* | *Disable*}	Enables or disables the Fan Status test.
fantac={*Enable* | *Disable*}	Enables or disables the Fan Tachometer test.
frupres={*Enable* | *Disable*}	Enables or disables the FRU Presence test.
fruidchksum={*Enable* | *Disable*}	Enables or disables the FRU ID Checksum test.
i2ctemp={*Enable* | *Disable*}	Enables or disables the I2C Tempature test.
psupplystatus={*Enable* | *Disable*}	Enable or disables the Power Supply Status test.

TABLE 11-5 `dmcatest` Command-Line Syntax

Argument	Description
powersupply={*Enable* | *Disable*}	Enable or disables the Power Supply On/Off test. The default is Disable.
dmca_ip=*IP_Address*	Required DMCA IP address.
dmcs_ip=*IP_Address*	Required DMCS IP address.
dev={dmc-active | dmc-standby}	Specifies dmc-active or dmc-standby.

`dmcstest` Command-Line Syntax

/opt/SUNWvtshm/bin/dmctest -o option=*value*, **enet=**{*Enable*\| *Disable*}, **epattype=**{*Seq+Rand*}, **epkts=**{*NUMERIC*\|*1,5*}, **target=**{*IP_Address*}, **etest=**{*Internal+External+Ping+Phy*}, **flash=**{*Enable* \| *Disable*}**, aport=**{*Enable* \| *Disable*}, **anum=**{*0+1+2+3*}, **a0on**={*On*\|*Off*\|*Toggle*}, **a1on**={*On*\|*Off*\|*Toggle*}, **a2on**={*On*\|*Off*\|*Toggle*}, **a3on**={*On*\|*Off*\|*Toggle*}, **health**=*{Enable* \| *Disable}*, **ipmi**={*Enable* \| *Disable*}, **rtc**={*Enable* \| *Disable*}, **fruidchksum**={*Enable* \| *Disable*}, **dmcs_ip**=*IP_Address*, **dev**={*dmc-active* \| *dmc-standby*}

TABLE 11-6 `dmcstest` Command-Line Syntax

Argument	Description
enet={*Enable*| *Disable*}	Enable or disables the Ethernet test.
epattype={*Seq+Rand*}	Indicates pattern type; Seq = Sequence, Rand = Random.
epkts={*NUMERIC*|*1,5*}	Indicates the number of packets to be tested.
target={*IP_Address*}	Required IP address for the Ethernet ping test.
etest={*Internal+External+Ping+Phy*}	Indicates the type of data to test.
flash={*Enable* | *Disable*}	Enable or disables the Flash test.
aport={*Enable* | *Disable*}	Enable or disables the ALARM port.
anum={*0+1+2+3*}	Indicates the alarm number.
a0on={*On*|*Off*|*Toggle*}	Toggles the `a0on` option on or off.
health=*{Enable* | *Disable}*	Enable or disables the Health test.
ipmi={*Enable* | *Disable*}	Enable or disables the IPMI test.
rtc={*Enable* | *Disable*}	Enable or disables the Real Time Clock test.

TABLE 11-6 `dmcstest` Command-Line Syntax

Argument	Description
fruidchksum={*Enable* | *Disable*}	Enable or Disable the FRU ID Checksum test
dmcs_ip=*IP_Address*	Required DMCS IP address
dev={*dmc-active* | *dmc-standby*}	Specifies whether the DMC card is active or standby.

CHAPTER 12

Sun Fire™ V880 FC-AL Disk Backplane Test (dpmtest)

dpmtest Description

dpmtest exercises and verifies the Fibre-Channel mass storage subsystem in Sun Fire V880 platforms. dpmtest exercises various tests in the Fibre-Channel Backplane firmware for validating the mass storage system.

No special hardware is required to run the dpmtest test.

dpmtest Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

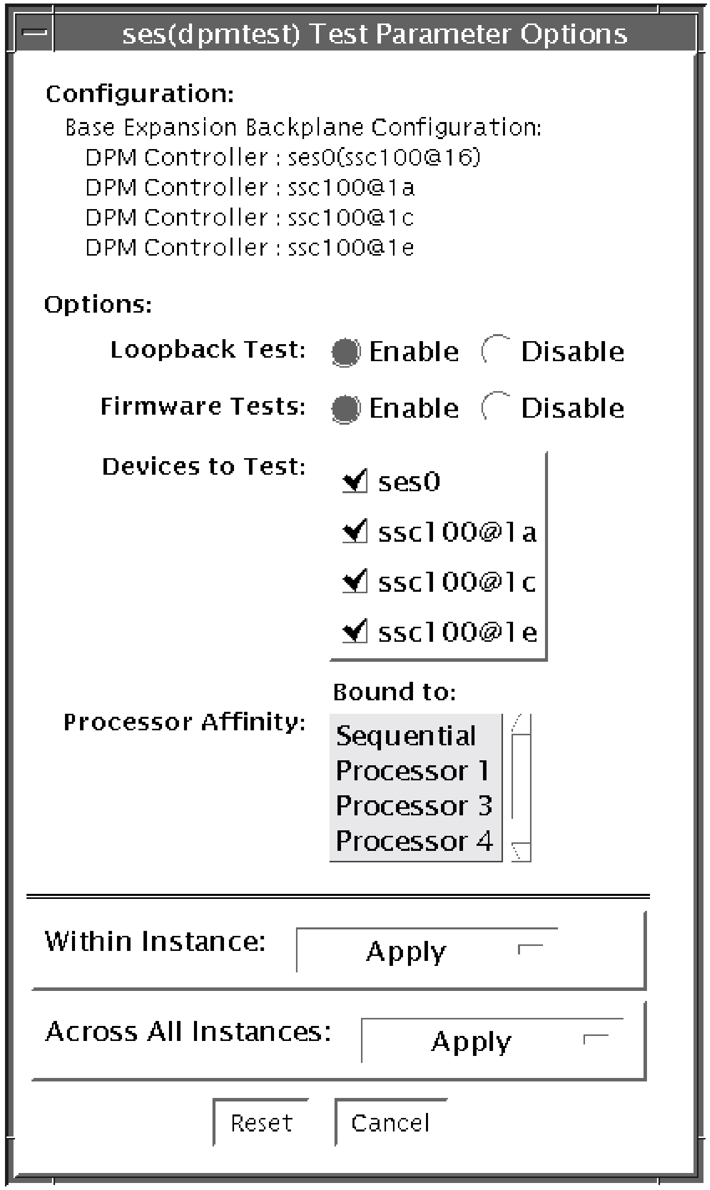


FIGURE 12-1 `dpmtest` Test Parameter Options Dialog Box

TABLE 12-1 `dpmtest` Test Options

Option	Description
Firmware Test	Runs the system-friendly firmware tests on each of the selected SES/SSC100 devices. By default this option is enabled.
Loopback Test	Causes the SES device to loop packets around the fiber loop with varying data patterns. The device reads the packet after the packet is received, and verifies that the data payload is correct. By default this option is enabled. Note: This test runs only on SES/SSC100 devices that are in the base backplane.
Devices to Test	The SES/SSC100 devices being tested. Users have an option to select or deselect each device for being tested. By default all the devices are selected for testing. Note: At least one device has to be selected for testing. If you try to deselect all of the devices, you will see an error message. Note: If the device has both fibre and I^2C paths, only the fibre path is listed under 'Devices to Test'. When you perform the tests on this device, the tests are run on both fibre and I^2C paths.

`dpmtest` Test Modes

TABLE 12-2 `dpmtest` Supported Test Modes

Test Mode	Description
Connection	The test opens each selected device, extracts information about the device (wwn/wwpn, firmware revision, drives installed, temperatures, and so on), and displays the information for the user. If the device has both fibre and I^2C paths, then information will be extracted from both the paths. After the test is performed on all the selected devices, the test closes the devices and exits.
Functional	The test opens each selected device and runs the selected subtests against the device. When fully run, the test closes the device and reports the results. Note: When no subtests are selected and you try to perform the functional testing, then just a configuration check will be performed.

`dpmtest` Command-Line Syntax

`/opt/SUNWvts/bin/dpmtest` *standard_arguments* **`-o dev=`***[device name]*, **`dpmdev=`***[device1+device2+...]*, **`fwtest=`***[Enable|Disable]*, **`lb=`***[Enable|Disable]*

TABLE 12-3 `dpmtest` Command-Line Syntax

Argument	Description
-o dev=*device-name*	*device-name* is the path name of the device being tested. The default value is `ses`. Since the current SunVTS infrastructure doesn't allow specifying multiple devices under the `dev` suboption, this suboption is not used in `dpmtest`. A new suboption `dpmdev` satisfies this requirement.
dpmdev= **[***device1+device2...]*	*device1*, *device2*,... represent the SES/SSC100 devices being tested. The default value is all the SSC100s present in the system. Note: The values for the `dpmdev` suboption can be device names such as `ses0`, `ses1`, `ssc100@16`, `ssc100@1a`, etc. Multiple values can be specified with a + (plus sign) seperator. An absolute path through fibre paths to devices are allowed (for example, `/dev/es/ses0`) as `dpmdev` suboption values. However, absolute paths through a I^2C path to devices are not allowed because commas are not allowed as part of a suboption value. Commas delimit suboptions in the options string (for example, `/devices/pci@9,700000/ebus@1/i2c@1,30/controller@0,16:ssc100`).

TABLE 12-3 `dpmtest` Command-Line Syntax *(Continued)*

Argument	Description
	Note: The following devices may be specified for the `dpmdev` suboption values in the Sun Fire V880 platforms: Fibre path: • `ses0` - Fibre path to base backplane's SSC100 (`/dev/es/ses0`) device on loopA. • `ses1` - Fibre path to base backplane's SSC100 (`/dev/es/ses1`) device on loopB. This is valid only when a PCI FC Network Adapter is connected to loopB. I^2C Path: • `ssc100@16` - Base backplane's SSC100 device on loopA through a `i2c` path. • `ssc1001a` - Base backplane's SSC100 device on loopB through a `i2c` path. • `ssc1001e` - Expansion backplane's SSC100 device on loopB through a `i2c` path. Note: The exact fibre path device node names (`ses0`, `ses1`, etc..) may vary depending on device nodes created in the system. The valid fibre path device nodes, that `dpmtest` found during probing, can be found under Devices to Test in the `dpmtest` Test Parameter Options dialog box.
`lb=`*[Enable|Disable]*	Enables or disables the loopback test. The default is Enable. Note: The loopback test runs only on SES/SSC100 devices that are in the base backplane.
`fwtest=`*[Enable|Disable]*	Enables or disables the firmware tests. The default value is Enable.

CHAPTER 13

Data Translation Look-Aside Buffer (dtlbtest)

dtlbtest Description

The dtlbtest verifies the proper functioning of DTLBs, but it does not measure performance of DTLBs.

Enhanced for UltraSPARC T1 (high-end processor with chip multithreading [CMT]) based systems.

Note – All the subtests/marches that are not intended for a particular platform are treated as an invalid option and the test provides a FATAL message indicating the invalid option.

You can now specify any march any number of times and in any order. The order of the entire command-line interface sequence is maintained for execution. This allows for a reduction in run time and provides an option for running an effective and stressful march earlier.

This test verifies the following function of DTLBs:

- Hit/miss tests – Stress DTLBs by generating large numbers of TLB hits and misses.
- Address Pattern tests – Verify that DTLBs correctly translate virtual addresses to physical addresses with stressful address patterns.

`dtlbtest` detects the installed CPU type and iterprets the CPU architecture.

`dtlbtest` is classified as an exclusive SunVTS test because its accuracy and coverage depend significantly on the system's background *quietness*. For the best result, avoid performing any other SunVTS tests in the background, and also disable all user processes and nonessential daemons.

Note – To test 4 MByte pages, Intimate Shared Memory (ISM) must be enabled. If not already present, append the following to `/etc/system` and reboot:

```
set shmsys:shminfo_shmmax=0xFFFFFFFFFFFFFFFF
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=10
```

`dtlbtest` Options

To reach the following dialog box, select the exclusive test mode and 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.

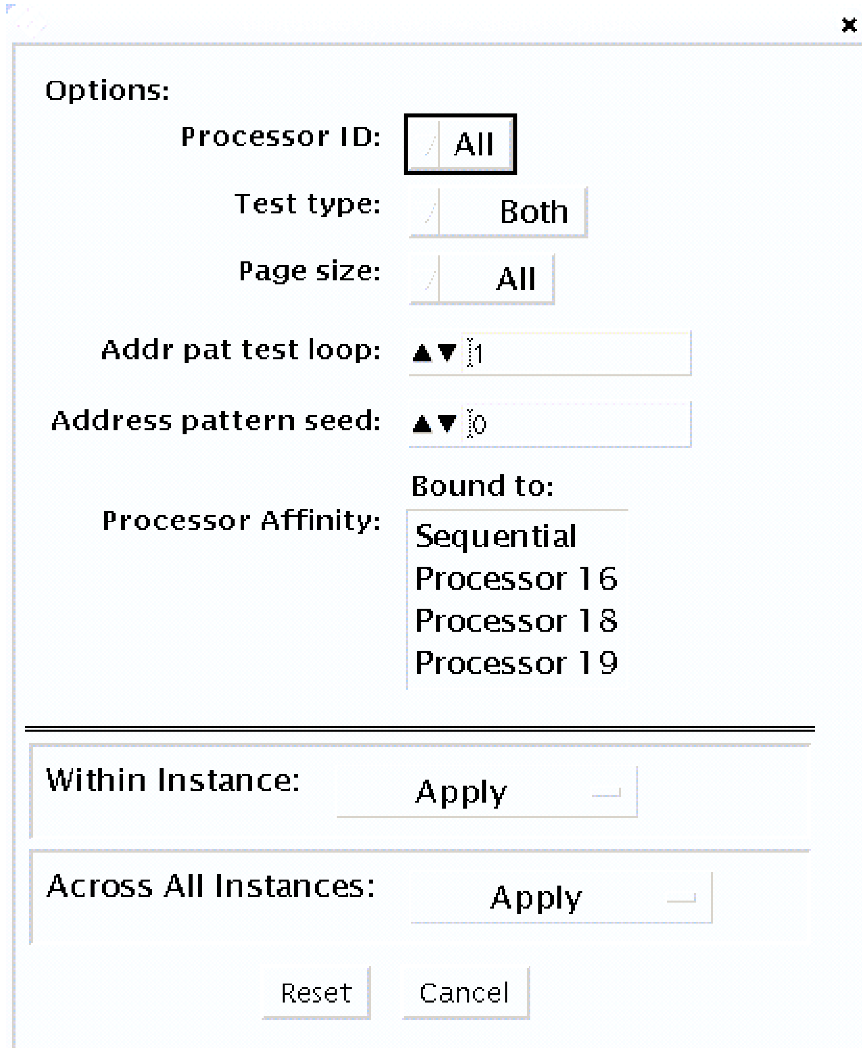


FIGURE 13-1 `dtlbtest` Test Parameter Options Dialog Box

TABLE 13-1 dtlbtest Options

Option	Description
Processor ID	Specifies the CPU ID for testing. The default value is all.
Test Type	Specifies the type of test need to be performed, Hit/miss, address pattern, or both. The default value is both.
Page Size	Specifies the DTLB page size to test. The default value is all. Possible page size values are 8k, 64k, 512K, and 4M.
Address Pattern loop count	Specifies the address pattern loop count. The default value is 1. Loop count value ranges between 1 and 1000.
Address pattern seed value	Specifies the address pattern seed value. The default value is 0.

TABLE 13-2 `dtlbtest` Supported Test Modes

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

`dtlbtest` Command-Line Syntax

/opt/SUNWvts/bin/sparcv9/dtlbtest [-scruvdtlxnf] [-p n] [-i n] [-w n] [-o cpu=*N*, **test=***x*, **psize=***x*, **aloop=***n*, **aseed=***n*]

TABLE 13-3 `dtlbtest` Command-Line Syntax

Argument	Description
cpu=*cpuid*	Specifies the CPU ID. The default value is all.
test=*testType*	Specifies the type of test h, a, or both. The default value is both.
psize=*pageSize*	Specifies the page size to be tested. The default value is all.
aloop=*loopCount*	Specifies the address pattern loop count. The default value is 1.
aseed=*seedValue*	Specifies the address pattern seed value. The default value is 0.

CHAPTER 14

Emulex HBA Test (emlxtest)

emlxtest verifies the functionality of the Emulex LP10000 Dual and Single Port 2 Gigabyte Fibre Channel PCI host bus adapters (HBAs). The supported HBAs are 133 MHz PCI-X 2 Gigabyte Fibre Channel based on the Emulex LP10000-S (Amber-2r) and LP10000DC-S (Crystal-2r) HBAs, which are the single and dual channel versions of the Emulex LP10000 line of products.

Note – emlxtest must be performed as a standalone test. Ensure that no other tests are running at the same time as the emlxtest.

---

emlxtest Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

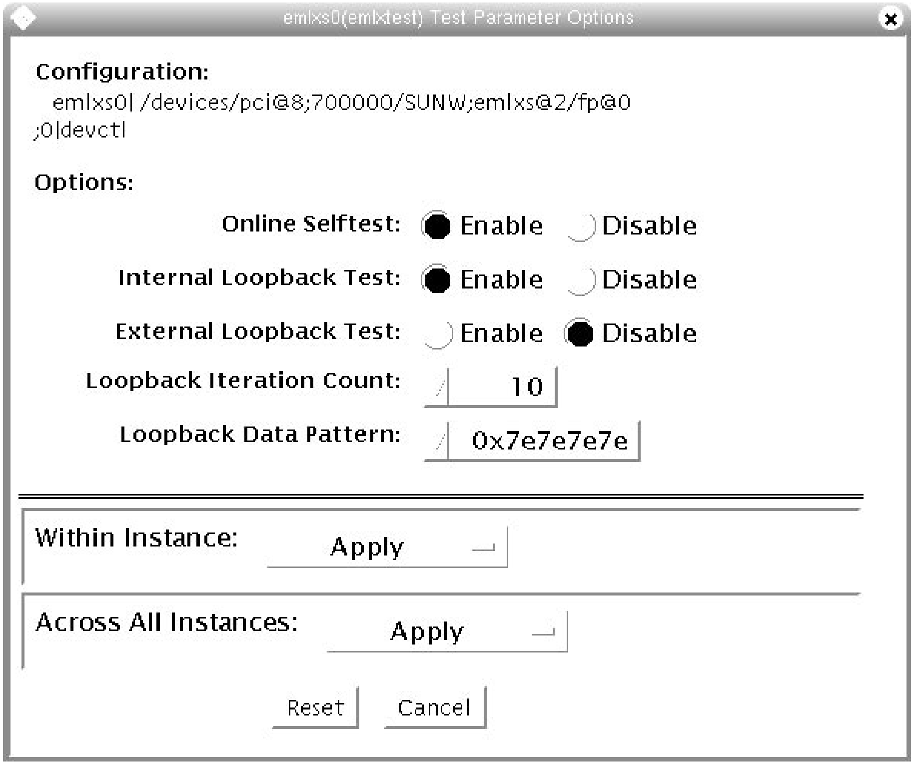


FIGURE 14-1 `emlxtest` Test Parameter Options Dialog Box

The following table lists the test parameter options for `emlxtest`.

TABLE 14-1 `emlxtest` Options

`emlxtest` **Options**	**Description**
Online Selftest	Evaluates the functionality of ISP hardware by performing the following tests: • Transmit FIFO test • Receive FIFO test • SRAM test • Misc. Register tests Run by default, but can be deselected.
Internal Loopback Test	Performs internal loopback test within the host adapter ISP hardware at the interface. This test is done with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected with the test parameters menu. Run by default, but can be deselected.
External Loopback Test	Performs an external loopback test. This test is performed with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected via the test parameters menu. This is an intervention test, because a loopback cable is needed from the transceiver to the receiver of the Emulex port when testing this port by itself. This subtest can also test the entire fibre channel loop when the loop is connected to the storage to be tested. Not run by default, but can be selected.
Loopback Iteration Count	Sets the number of times to loop the internal 10-bit, internal 1-bit, and external loopback tests. Default value is `10`.
Loopback Data Pattern	Selects the data pattern to loop for the internal 10-bit, internal 1-bit, and external loopback tests. Default value is `0x7e7e7e7e`.

`emlxtest` Test Modes

TABLE 14-2 `emlxtest` Supported Test Modes

Test Mode	Description
Connection	Opens and closes the Emulex port.
Exclusive	Performs the full set of tests.

`emlxtest` Command-Line Syntax

`/opt/SUNWvts/bin/emlxtest` *standard-arguments*
`-x -v -o dev=`*device-name***`,run-connect=`***Yes* | *No***`,selftest=`**
Enable | *Disable* , **`ilb=`***Enable* | *Disable* , **`elb=`***Enable* | *Disable* , **`lbfcount=`***iteration-count* ,
`lbfpattern=`*hex-pattern*

Note – The `-x` option must be used from the command line.

TABLE 14-3 `emlxtest` Command-Line Syntax

Argument	Description
`dev`=*device-name*	The name of the device to test.
`selftest=`*Enable* | *Disable*	Enables or disables the `selftest` command. Evaluates the functionality of the ISP hardware. Enabled by default.
`ilb=`*Enable* | *Disable*	Enables or disables the internal test. Performs internal loopback test within the host adapter ISP hardware at the 1-bit interface. Enabled by default.
`elb=`*Enable* | *Disable*	Enables or disables the external loopback test. The desired data pattern, transfer length, and iteration count can be selected on the test parameters menu. Requires a cable for this intervention test. Disabled by default.
`lbfcount=`*iteration-count*	Controls the number of times the loopback test will run, for example, 100. Default value is 10.
`lbfpattern=`*hex-pattern*	Lists the data pattern to loop, for example, 0x7E7E7E7E. Default value is 0x7E7E7E7E.

CHAPTER 15

Environmental Test (envtest)

envtest Description

envtest exercises the I^2C bus on the Sun Enterprise™ 450 product line. envtest contains five subtests to test and report on the power supply status, system temperature status, fan speed, disk LEDs, and front panel and keyswitch.

envtest is not scalable.

Note – Do not run envtest while the system is under a heavy load or false failures may be reported.

---

envtest Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

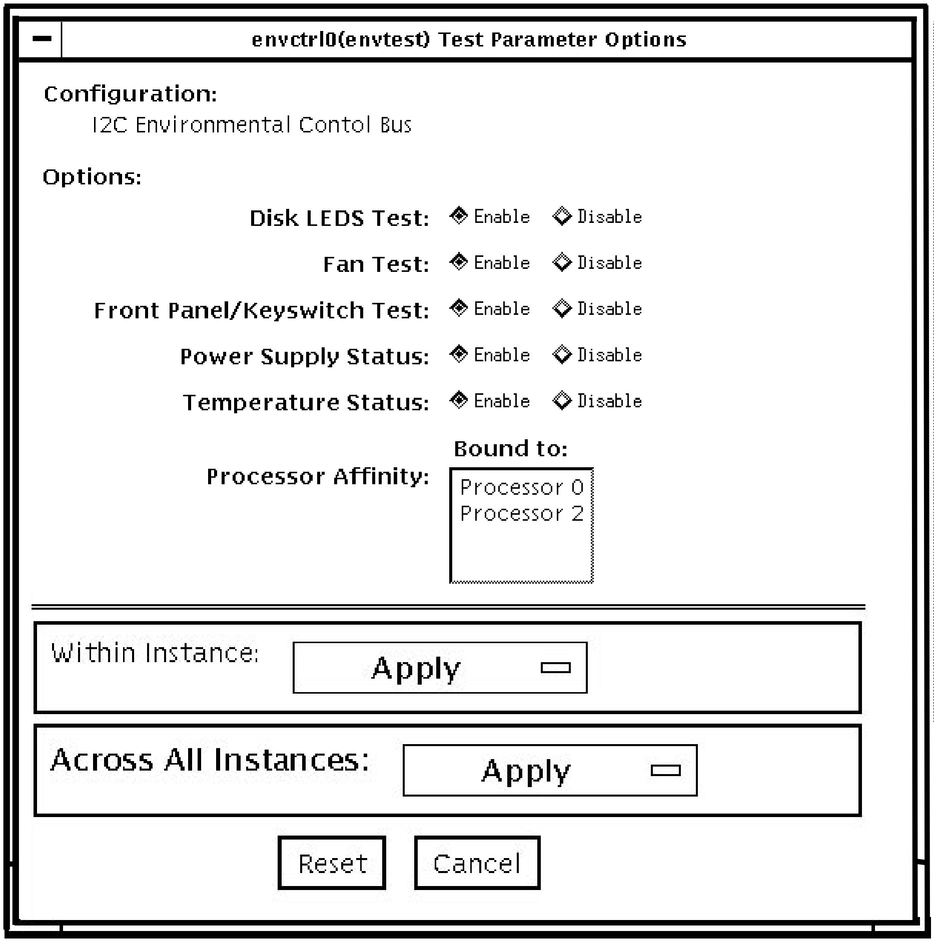


FIGURE 15-1 `envtest` Test Parameter Options Dialog Box

TABLE 15-1 `envtest` Options

`envtest` **Options**	**Description**
Disk LEDs test	Illuminates each LED on the disk backplane(s) to green, then amber, and then back to the original state. The test then illuminates all disk LEDs to green, then amber, and then back to their original state. This test is only enabled in Functional test mode.
Fan test	Cycles each fanbank speed to low, medium, and high, then verifies the correct speed. Next, each fanbank is stopped, one at a time. The test then verifies that a fan fault has occurred. Next, the watchdog timer is invoked to simulate a catastrophic failure. The test verifies that the system set all fanbanks to high and then resets the fan speed to normal. This test is only enabled in Functional test.
Front Panel and Keyswitch test	Flashes each individual LED on the front panel to on (green or amber), then off, and then back to the original state. The test then illuminates all front panel LEDs then sets them back to their original state. The Power On LED is read-only and will not be cycled. The test then displays the current keyswitch position. This test is only enabled in Functional test mode.
Power Supply Status	Identifies the number of power supplies that are in the system, and the state of each power supply, and verifies that the power supply temperatures are within normal operating parameters. This test is enabled in all modes.
Temperature Status	Identifies the current temperature of each CPU in the system, and the ambient temperature of the system. `envtest` verifies that all temperatures are within normal operating parameters. This test is enabled in all modes.

`envtest` Test Modes

`envtest` supports Connection and Functional test modes.

TABLE 15-2 `envtest` Supported Test Modes

Test Mode	**Description**
Connection	Reports the status of the power supplies, the temperature sensors within the system, and verifies normal operating parameters.
Functional	Tests the disk back panel, front panel LEDs, and fan control circuitry. Uses the same functionality as online Functional mode and connection mode

`envtest` Command-Line Syntax

`/opt/SUNWvts/bin/envtest` [*standard arguments*]
`-o dev=`*raw-device-name***`,diskleds=`***E/D***`,fans`**=*E/D***`,fpanel`**=*E/D***`,`**
`psupply`=*E/D***`,temp`**=*E/D*

TABLE 15-3 `envtest` Command-Line Syntax

Argument	Description
`dev=`*raw-device-name*	Specifies the name of the raw device to test.
`diskleds=`*enable|disable*	Enables or disables the Disk LEDS test.
`fans=`*enable|disable*	Enables or disables the Fans test.
`fpanel=`*enable|disable*	Enables or disables the Front Panel test.
`psupply=`*enable|disable*	Enables or disables the Power Supply test
`temp=`*enable|disable*	Enables or disables the Temperature test.

CHAPTER 16

Environmental Test (env2test)

env2test Description

env2test exercises and validates the I^2C bus on the Sun Enterprise 250 systems.

Five subtests in env2test test and report the status of the power supply, system temperature, fan speed, disk LEDs, front panel, and keyswitch.

env2test is not scalable.

Note – Do not run env2test while the system is under a heavy load. False failures might be reported.

---

env2test Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

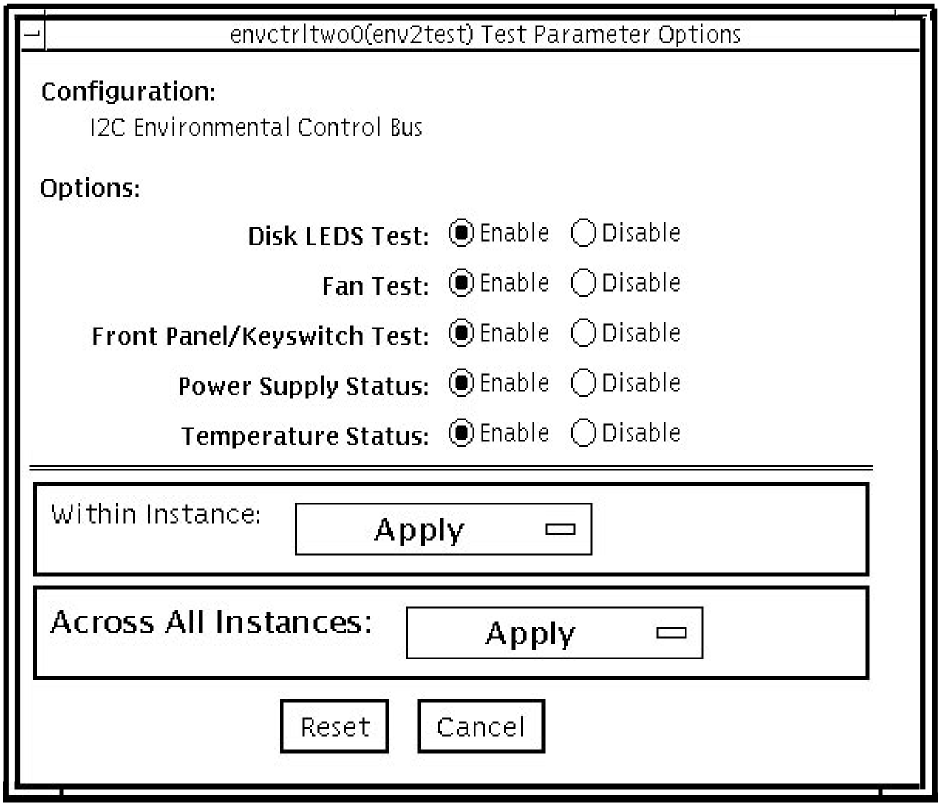


FIGURE 16-1 `env2test` Test Parameter Options Dialog Box

TABLE 16-1 `env2test` Options

`env2test` **Options**	**Description**
Disk LEDs test	Determines the number of disks that are present and the state of each disk. Illuminates each LED on the disk backplane to amber, and then back to yhe original state. The test then illuminates all disk LEDs to amber, and then back to the original state. This test is only enabled in Functional mode.
Fan test	Sets the fanbank speed to halfway between the current speed and max speed, then verifies the correct speed. Next, the fanbank speed is set to max speed, and the speed is verified. The fanbank speeds are then returned to normal, and then verified. This test is only enabled in Functional mode.
Front Panel and Keyswitch test	Flashes each individual LED on the front panel to on (green or amber), then off, and then back to the original state. The test then illuminates all front panel LEDs then sets them back to their original state. The Power On LED is read-only and is not cycled. The test then displays the current keyswitch position. This test is only enabled in Functional mode.
Power Supply Status	Identifies the number of power supplies that are in the system and the state of each power supply. This test is enabled in all modes.
Temperature Status	Identifies the current temperature of each CPU in the system, the ambient temperatures of the system, the temperature on the SCSI and power distribution boards, and verifies that all temperatures are within normal operating parameters. This test is enabled in all modes.

`env2test` Test Modes

`env2test` supports Connection and Functional tests as described in the table below.

TABLE 16-2 `env2test` Supported Test Modes

Test Mode	**Description**
Connection	Reports the status of the power supplies and the temperature sensors within the system, and verifies normal operating parameters.
Functional	Tests the disk back panel, front panel LEDs, and fan control circuitry. Uses the same functionality as online mode and connection mode.

`env2test` Command-Line Syntax

/opt/SUNWvts/bin/env2test [*standard arguments*]
-o dev=*device_name***,diskleds=***E/D***,env_mon=***poll_interval***,fans=***E/D,*
fpanel=*E/D***,psupply=***E/D***,temp=***E/D*

TABLE 16-3 `env2test` Command-Line Syntax

Argument	Description
dev=*raw_device_name*	Specifies the name of the raw device to test.
diskleds=*enable|disable*	Enables or disables diskleds test.
env_mon=*poll_interval*	Displays all system environmental statics every `poll_interval` seconds. (Display only, does not test.)
fans=*enable|disable*	Enables or disables the Fan test.
fpanel=*enable|disable*	Enables or disables the Front Panel test.
psupply=*enable|disable*	Enables or disables the Power Supply test.
temp=*enable|disable*	Enables or disables the Temperature test.

CHAPTER 17

Environmental Test (env3test)

env3test Description

env3test is an environmental control test for the Sun Blade 1000/2000, Sun Blade 100/150 and Sun Blade 1500/2500, and A70 workstation systems. Although the central function of this test environmentals and thus common to all these platforms, the exact behavior of the test is platform-specific.

For Sun Blade 100/150 and 1000/2000 workstayions, the test monitors the system by reading temperatures and fan speeds, as well as their limits. The test reports whether the temperatures and fan speeds fall within system environmental condition limits. The MAX 1617 temperature sensor maintains a set of hard limits for the maximum and minimum temperatures allowed within the system. If the temperature passes one of these limits, the system performs a hard shutdown to protect hardware components. env3test also checks against the environmental monitor daemon, which maintains its own limits. By reading these temperature values, the test reports the possibility of a failure of the system environmental feedback loop between the fans and the temperature sensors.

For Sun Blade 1500 and 2500 systems, the test displays the temperature sensors and temperatures, and all the fans and their speeds in RPM. The test has an option for logging this information. The test will fail if one or more fans are bad, are disconnected or are weak. An appropriate error message indicates a disconnected or bad case, or a weak fan case.

For the A70, the test displays the temperature sensors and temperatures, and all the fans and their speeds in RPM. It has an option for logging this information. The test fails if the fan status indicates a fan fault.

For Sun Blade 1500 and 2500, and the A70 workstations, the `env3test` supports the Exclusive and Connection test modes.

Note – On Sun Blade 100/150, 1000/2000, if `env3test` fails to register temperature values, the system temperature indicators could be faulty.

Note – On Sun Blade 1500/2500 workstations, `env3test` fails if a fan is bad, disconnected, or weak. An error message indicates which case. For the Sun Blade 1500, the outtake-fan RPM cannot be monitored due to hardware limitation.

Note – On Sun Blade 100/150, 1000/2000 workstaions, `env3test` requires Solaris 8 10/00 OS or later.

`env3test` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

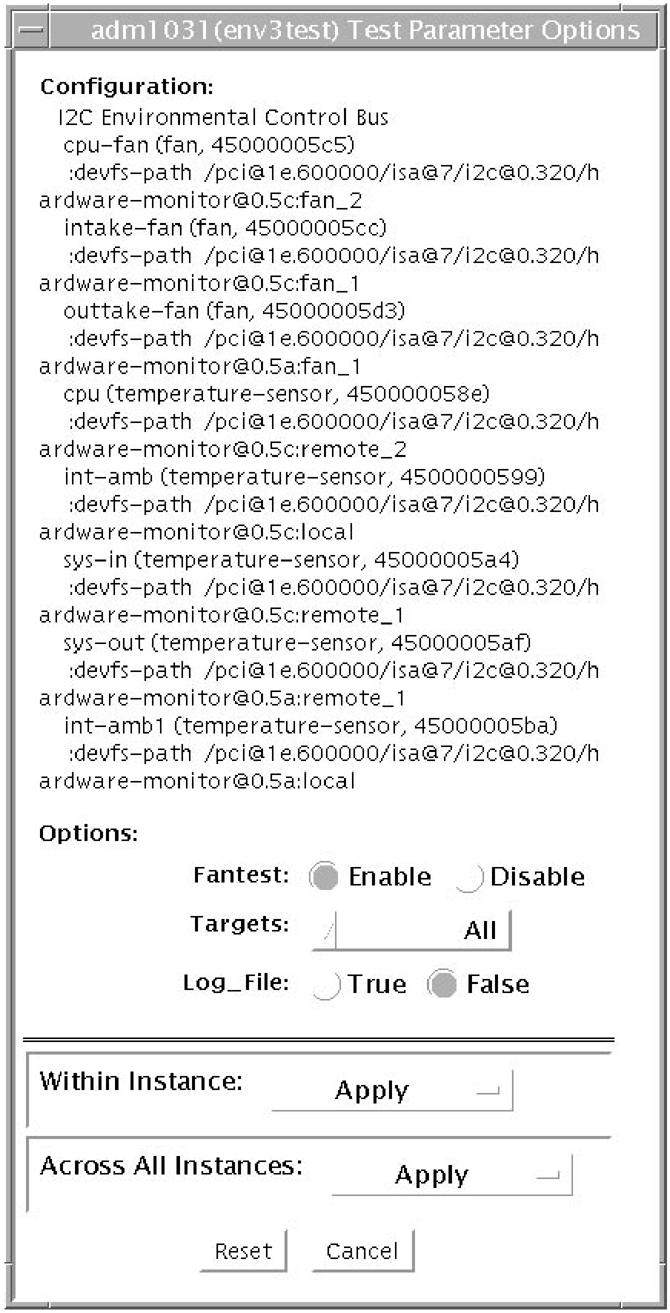


FIGURE 17-1 `env3test` Test Parameter Options Dialog Box

TABLE 17-1 `env3test` Options

Option	Description
Fan Test	Displays the percentage of performance at which each fan is running. The temperature values are in degrees Celsius.
Target	Specifies the target devices.

Log File	Logs two lines of information read from the system into the log file `/var/opt/SUNWvts/logs/env3test.log`. The first line is a time stamp. The second line is a list of names and the read values, for example,
	`Wed May 24 13:55:57 2000`
	`system-fan , 19, cpu-fan , 49, power-supply-fan , 100, cpu , 81, cpu-ambient ,24`

env3test Test Modes

TABLE 17-2 env3test Supported Test Modes

Test Mode	Description
Connection	Attempts connection to the device. Supported on Sun Blade 1500/2500 and A70 workstations.
Functional	Reports the received information to the GUI logging window in verbose mode. Supported on Sun Blade 150/1000 workstations.
Exclusive	Reports the received information to the GUI logging window in verbose mode. Supported on Sun Blade 1500/2500 and A70 workstations.

env3test Command-Line Syntax

/opt/SUNWvts/bin/env3test [*standard arguments*] **-o dev=***driver_name***, logging=***true*|*false*, **fantest**=*enable*, **fan**=*All/intake-fan+outtake-fan*

TABLE 17-3 env3test Command-Line Syntax

Argument	Description
dev=*driver-name*	Specifies the driver name for this test. The driver for the Sun Blade 1000 workstation is `max1617`.
logging=*true*|*false*	Enables or disables the logging feature.
fantest=*enable*	Enables the fan test.
fan=*All/intake-fan+outtake-fan*	Specifies the fan to be tested (such as, `cpu-fan`, `intake-fan`, or `outtake-fan`). The keyword All specifies all the fans `env3test` probe found.

CHAPTER 18

Environmental Test (env5test)

env5test Description

env5test exercises and validates the environmental subsytems of the Sun Fire 280R product line. This test contains subtests to exercise a system's fans, keyswitch, LEDs, power supplies and temperature sensors.

This test is not scalable.

Note – Only the 64-bit version of this test is supported.

---

env5test Test Requirements

You must install the SUNWpiclh, SUNWpiclr, SUNWpiclu, and SUNWpiclx picl packages correctly before running env5test.

Verify that the `picld` daemon is running by typing the following:

```
# ps -ef |grep picld
root 100077      1  0   Sep 11 ?       23:40 /usr/lib/picl/picld
```

If the daemon is not running, run the script to restart it by typing the following:

```
# /etc/init.d/picld start
```

env5test Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

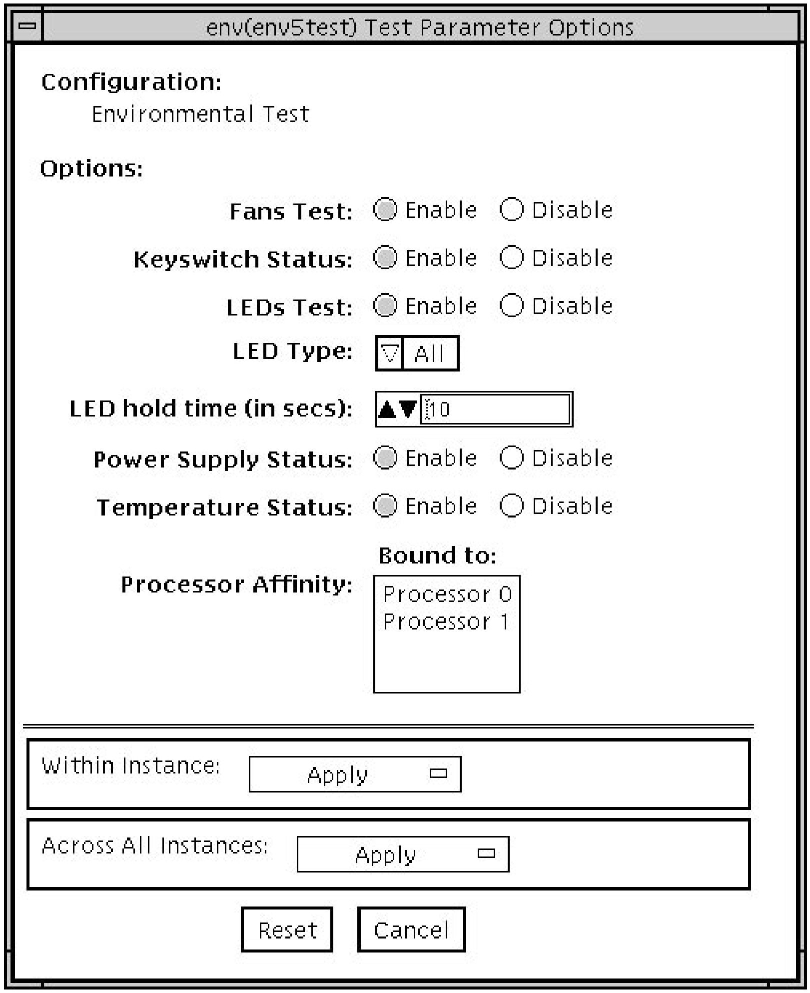


FIGURE 18-1 `env5test` Test Parameter Options Dialog Box

TABLE 18-1 `env5test` Options

Options	Description
Fan Test	Cycles each fan bank in the system and identifies its current speed and state. Displays fault information if the state is not correct. Provides different test coverage for various types of fans based on their properties. Only enabled in Functional test mode.
Keyswitch Status	Displays the current keyswitch position. Enabled in all modes.
Leds Test	Enables or disables the LEDS subtest. Default is Disable. Flashes each individual green or amber LED in the system on, then off, then back to its original state. You can select to test all LEDs (by default) or test by categories. Also specifies how long the LEDs stay on during the exercise. Only enabled in Functional test mode.
Led Type	Sets the type of LED category to be tested. Default is All.
Led Hold Time	Sets the number of seconds that LEDs are turned on during the LED subtest. Values are 0 to 10. Default is 0.
Power Supply Status	Identifies the number of power supplies in the system, the state of each one, and current (I) draw information if applicable. Displays fault information if the state of the power supply is incorrect. Enabled in all modes.
Temperature Status	Identifies the current temperature of temperature sensors in the system and verifies that all temperatures are within normal operating parameters. Enabled in all modes.

`env5test` Test Modes

TABLE 18-2 `env5test` Supported Test Modes

Test Mode	Description
Connection	Runs the Keyswitch, Power Supply, and Temperature subtests. Reports on the status only of system fans. Verifies normal operating parameters.
Functional	Runs the full set of subtests.

`env5test` Command-Line Syntax

/opt/SUNWvts/bin/sparcv9/env5test *standard-arguments*
-o dev=*raw-device-name***,leds=***E*|*D***,ledtype=***category***,ledtime=***num-seconds***,**
keys=*E*|*D***,fans=***E*|*D***,psupply=***E*|*D***,temp=***E*|*D*

TABLE 18-3 `env5test` Command-Line Syntax

Argument	Description
dev=*raw-device-name*	Specifies the name of the raw device to test. Default is `/dev/env`.
leds=*E*|*D*	Enables or disables the LEDS subtest. Default is Disable.
ledtype=*category*	Sets the type of LED category to be tested. Default is All.
ledtime=*num-seconds*	Sets the number of seconds that LEDs are turned on during the LED subtest. Values are 0 to 10. Default is 0.
keys=*E*|*D*	Enables or disables the Keyswitch subtest. Default is Enable.
fans=*E*|*D*	Enables or disables the Fans subtest. Default is Disable.
psupply=*E*|*D*	Enables or disables the Power supply subtest. Default is Enable.
temp=*E*|*D*	Enables or disables the Temperature subtest. Default is Enable.

CHAPTER 19

Environmental Test (env6test)

env6test Description

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 Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

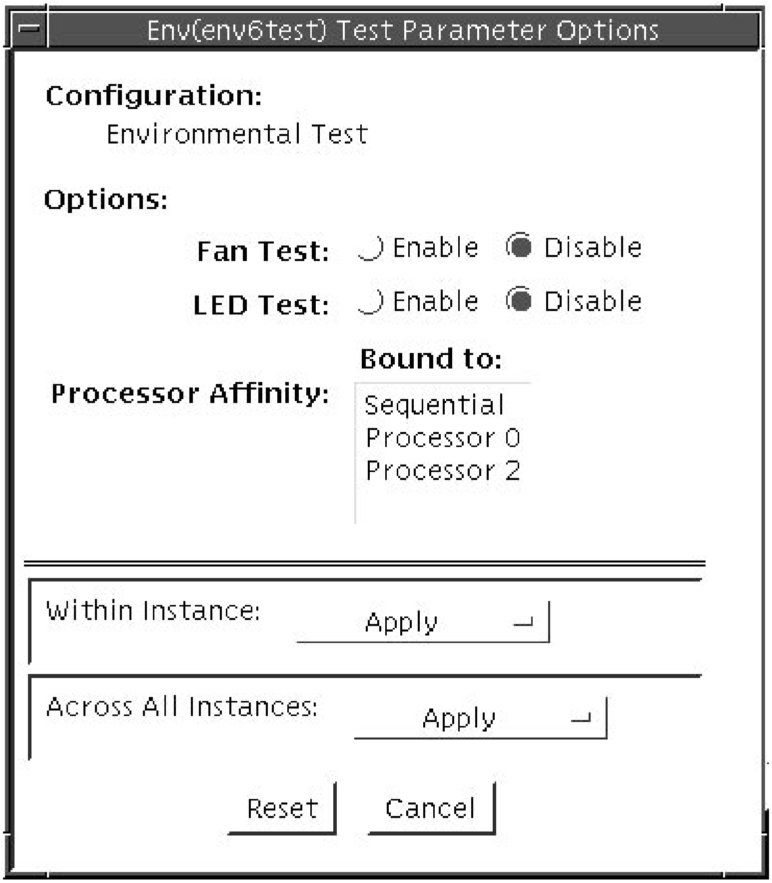


FIGURE 19-1 `env6test` Test Parameter Options Dialog Box

TABLE 19-1 `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 19-2 `env6test` Supported Test Modes

Test Mode	Description
Connection	Reports current state of devices.
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 19-3 `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. The default value is Disable.
`fan=`*Enable* | *Disable*	Enables or disables the Fans subtest. The default value is Disable.

CHAPTER 20

Floating Point Unit Test (fputest)

fputest Description

The fputest checks the functionality of the floating point unit in a Sun SPARC based CPU. The test verifies the functionality by various arithmetic operations. In addition, the fputest stresses the CPU with the use of benchmarks. Both single and double precision numbers are used for the operations.

When fputest is chosen in Exclusive test mode from the SunVTS graphical user interface, it may run multiple instances in parallel on different CPUs. The number of such instances that may be running in parallel at the same time is dynamically determined depending on system resources.

fputest Subtests

Instruction tests include:

- FSR Register test
- Registers test
- NACK test
- Move Registers test

- Positive to Negative test
- Negative to Positive test
- Absolute test
- Single-Precision Integer to Floating Point test
- Double-Precision Integer to Floating Point test
- Single-Precision Floating Point to Integer test
- Double-Precision Floating Point to Integer test
- Single-Precision Round Toward Zero test
- Double-Precision Round Toward Zero test
- Single to Double-Precision Format Conversion test
- Double to Single-Precision Format Conversion test
- Single and Double-Precision Addition, Subtraction, Multiplication, Square-root, Division, and Compare tests
- Single and Double-Precision Compare and Exception if Unordered tests
- Branching and No Branching on Condition Instructions tests
- Single and Double-Precision Chaining tests
- Weitek Status tests
- Lock test
- Single and Double-Precision Datapath tests
- Timing (load) test

Benchmark tests include:

- Lapack test
- Cparanoia test
- Kcsqrt test
- Kcdiv test
- Clorenz test
- Cvector test

`fputest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

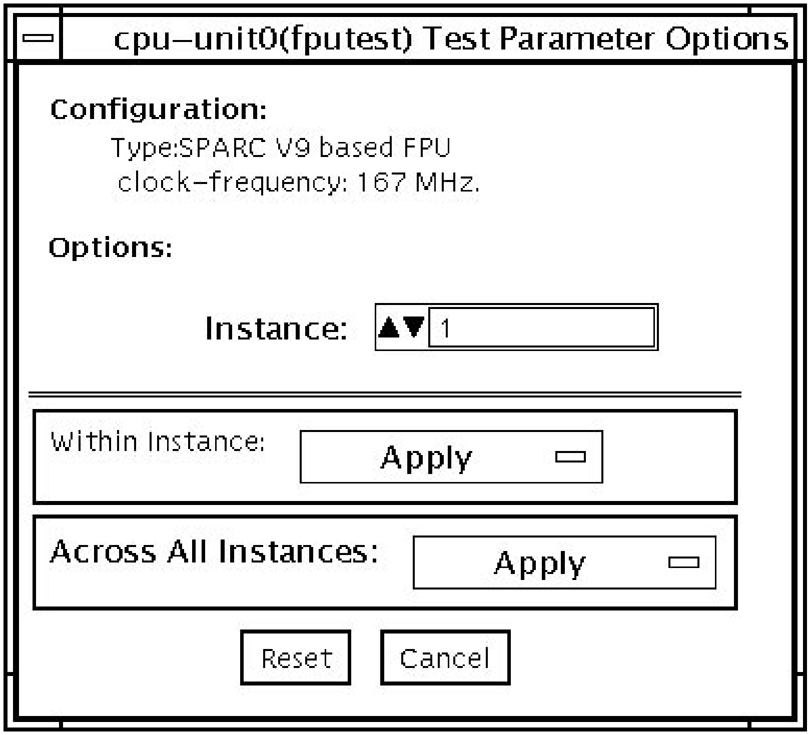


FIGURE 20-1 `fputest` Test Parameter Options Dialog Box

Note – It is not advisable to use the Processor Affinity option for this test. Doing so reduces the effectiveness of the test.

---

fputest Test Modes

TABLE 20-1 fputest Supported Test Modes

Test Mode	Description
Connection	Includes all the instruction tests.
Functional	Performs all the instruction tests and all the benchmark tests.
Exclusive	Performs several FPU benchmark tests.
Online	Supported.

fputest Command-Line Syntax

/opt/SUNWvts/bin/fputest [**-scruvdtlxnf**] [**-p n**] [**-i n**] [**-w n**] [**-o dev=cpu-unit***N*]

Note – Options -s, -p, -i, and -w are not applicable when tests are invoked from the command line.

TABLE 20-2 fputest Command-Line Syntax

Argument	Description
-s	Enables SunVTS mode.
-c	Enables core file.
-r	Enables run on error.
-u	Enables list usage.
-v	Enables Verbose mode.
-d	Enables Debug mode.
-t	Enables test function trace mode.
-l	Enables Online mode, and forks tests automatically against all online CPU IDs in the system.
-x	Enables Exlusive mode, and forks tests automatically against all online CPU IDs in the system.

TABLE 20-2 `fputest` Command-Line Syntax

Argument	Description
`-n`	Enables Connection mode.
`-f`	Enables Offline mode, and forks tests automatically against all online CPU IDs in the system.
-p n	*n* specifies the number of passes. The default is 1.
`-i` *n*	*n* specifies the number of total instances for the test. The default is 1.
`-w` *n*	*n* specifies which instance this test is assignes. The default is 0.
`-o`	Enables test specific command arguments.
`dev=cpu-unit`*N*	Specifies the CPU unit to be tested. *N* specifies the numeric ID of online CPU. This is not a required option.

Usage Examples:

Note – When using `fputest` specific arguments, the `-o` and `dev` options are required.

To execute exclusive `fputest` with verbose messages against CPU 72, use the following syntax:

```
# /opt/SUNWvts/bin/fputest -xvo dev=cpu-unit72
```

To execute functional `fputest` with verbose messages against CPU 3, use the following syntax:

```
# /opt/SUNWvts/bin/fputest -fvo dev=cpu-unit3
```

CHAPTER 21

IEEE 1394 Camera Test (`fwcamtest`)

`fwcamtest` Description

`fwcamtest` tests the parameters and display functions of an IEEE 1394 digital camera, such as vid mode, frame rate, and frames received. This test currently supports the LG PC-10 camera.

Note – Do not run the `fwcamtest` and `vmemtest` at the same time on any Sun Blade system. This might cause the tests to fail.

Note – Do not start Sun Forum™ (or any application that uses the `dcam0` device) while `fwcamtest` is running. This causes the test to fail.

---

fwcamtest Test Requirements

This section describes the fwcamtest test requirements.

Setting 24-Bit Depth in the Window Environment

The system that runs fwcamtest must already be running a window environment, such as JDS. If the system has no windowing environment, or is only displaying the login window, fwcamtest will neither pass nor fail.

Note – Your window system must be operating in 24-bit depth to run the display test. Instructions for changing this setting are below.

If you are working in JDS, you can change your system to 24-bit depth by editing the file /usr/dt/config/Xservers or /etc/dt/config/Xservers. The file /etc/dt/config/Xservers overrides the file /usr/dt/config/Xservers. Edit the appropriate file to include the following line:

```
:0   Local local_uid@console root /usr/openwin/bin/Xsun :0
-nobanner -dev /dev/fbs/ffb0 defdepth 24 defclass TrueColor
```

Testing Through a Remote Connection

While running fwcamtest through a remote connection (such as a Telnet session), if the DISPLAY variable is not set properly, it will cause numerous warning messages to display. These messages are logged and can fill up the log files. To avoid this, set your DISPLAY variable for the local host, and perform xhost *remote-host* on the local host before you start SunVTS and run fwcamtest.

fwcamtest Subtests

TABLE 21-1 describes the `fwcamtest` subtests.

TABLE 21-1 `fwcamtest` Subtests

Subtest	Description
Parameter test	Tests the digital camera parameters such as vid mode and brightness.
Framereceive test	Initializes the vid mode, frame rate and ring buffer capacity parameters, then checks for the frame received.
Display test	Displays the captured frames. This display test sets up the 1394 bus for asynchronous transfer mode. The display test only displays the frames on the host running the test. The test cannot display on a remote host.

fwcamtest Options

To reach the following 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

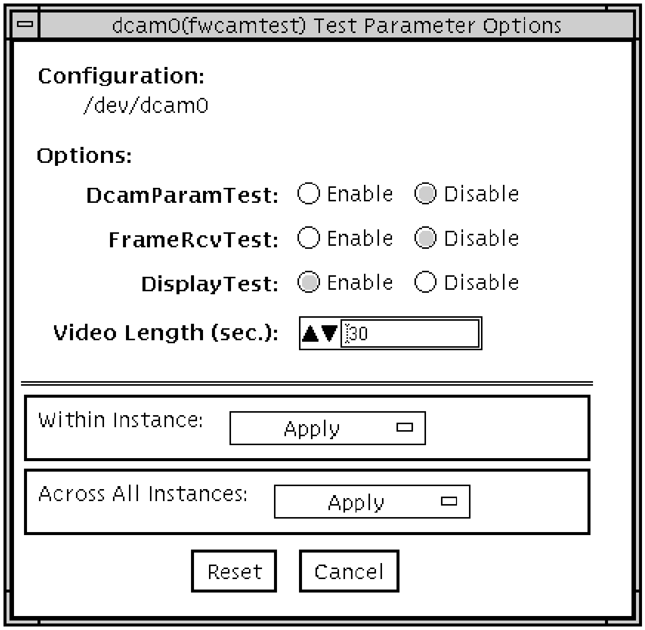


FIGURE 21-1 `fwcamtest` Test Parameter Options Dialog Box

TABLE 21-2 `fwcamtest` Options

`fwcamtest` **Options**	**Description**
DcamParamTest	Enables or disables the Parameter subtest.
FrameRevTest	Enables or disables the Frame Receive subtest.
DisplayTest	Enables or disables the Display subtest.
Video Length	Determines, in seconds, how long the screen display lasts.

`fwcamtest` Test Modes

TABLE 21-3 `fwcamtest` Supported Test Modes

Test Mode	Description
Functional (Offline)	Runs the full set of tests.

`fwcamtest` Command-Line Syntax

/opt/SUNWvts/bin/fwcamtest *standard-arguments*
-o dev=*dcam0***,dcamparam=***E(nable)*|*D(isable)***,framercv=***E*|*D***,display=***E*|*D***,**
T=*seconds*

TABLE 21-4 `fwcamtest` Command-Line Syntax

Argument	Description
dev=*dcam0*	Specifies the device name for this test.
dcamparam=*E(nable)*|*D(isable)*	Enables or disables the Parameter subtest.
framercv=*E*|*D*	Enables or disables the Frame Receive subtest.
display=*E*|*D*	Enables or disables the Display subtest.
T=*seconds*	Specifies the time period of the display test in seconds.

CHAPTER 22

Sun XVR-2500 Graphics Accelerator Test (graphicstest)

graphicstest Description

graphicstest verifies the functionality of XVR-2500 frame buffers. 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 frame buffer's accelerator port while running graphicstest. This combination causes SunVTS to return incorrect errors.

graphicstest Test Requirements

To run graphicstest as super user, you must use the console device or start the desktop with the -ac option.

Note – The CDE desktop is an X client that prevents the X server from restarting after each `graphicstest` finishes testing the kfb device.

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

```
# xset -dpms
```

The display resolution must be 1280x1024. To change resolution, go to a UNIX prompt and type:

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

For full instructions on testing frame buffers, see "Testing Frame Buffers" on page 11.

`graphicstest` Test Preparation

You must run `graphicstest` through the SunVTS user interface. The desktop must be running on the XVR-2500, and the desktop must be setup to run in 24-bit mode (the default depth for XVR-2500). Perform on of the following procedures prior to performing `graphicstest` to ensure that the test runs as smoothly as possible.

▼ To Run `graphicstest` in a Desktop GUI

1. **Turn Power Management off, if it is enabled. See** "`graphicstest` Test Requirements" on page 151**.**
2. **Verify that no other program is running that might modify the screen during the test.**
3. **Verify that you have permission to lock the X server. `graphicstest` is designed to lock the X server during testing to prevent screen changes.**

4. **Verify that the JDS login window is not displayed during testing.**
5. **For frame buffers other than kfb, verify that the desktop is running on one frame buffer only.**

   If a kfb device is in the platform, the desktop must not be run on any other non-kfb graphics devices in the system when performing SunVTS.

Note – To perform `graphicstest` on kfb frame buffers, the desktop must be running on each kfb device in the system.

`graphicstest` Options

To reach the following dialog box, right-click on the test name in the System Map and select Test Parameter Options. Because graphics test can test multiple types of frame buffers, the test name that is displayed will correspond to the particular frame buffer being tested. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system might not include the device appropriate to this test. Refer to the *SunVTS User's Guide*.

By default, all `graphicstest` options are enabled.

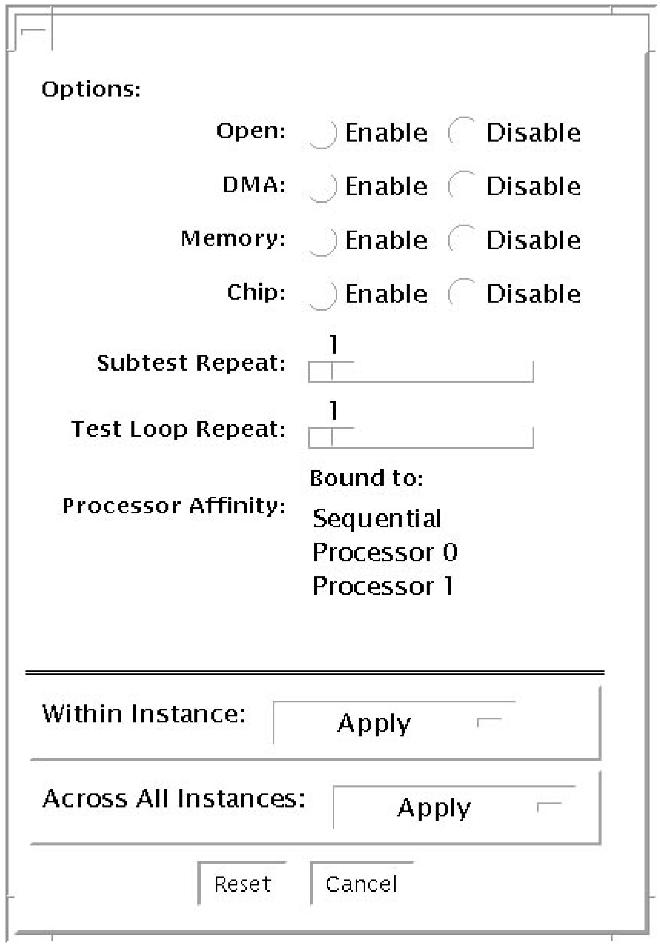


FIGURE 22-1 `graphicstest` kfb0 Test Parameter Options Dialog Box

TABLE 22-1 `graphicstest` kfb0 Options

Option	Description
Open test	Verifies that the graphics device can be correctly opened. First the device is opened with an open system call. Next the register and frame buffer regions are mapped into the virtual address space of the test process. Next a subset of the read/write resisters are written to and read back from to verify that the register region was correctly mapped. Next selected areas of the frame buffer are written to and read back from to see if the the frame buffer was correctly mapped. This test takes very little time and no progress is displayed.
Memory test	Verifies that all the memory on the graphics card is working. The test opens the device, allocates all the device memory, and then writes to memory using DMA or programmed I/O. The test then reads back the values from memory and checks to see if the correct values were read back.
DMA test	Verifies that the DMA engine on the device is functioning. The test writes data to an area of the frame buffer using DMA, then reads back the same area using programmed I/O and checks if the values are correct. The second part of the test writes data into the frame buffer and reads it back using the DMA engine, checking to see if the correct values were read back.
Chip Test	Exercises the graphics chip on the board to see if it is functioning correctly. The chip test exercises the main data paths through the graphics chip. Commands are sent to the graphics board to initiate a graphics program. Once the program has run the results are read from the frame buffer to verify if they are correct.

`graphicstest` Test Modes

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

TABLE 22-2 `graphicstest` Supported Test Modes

Test Mode	Description
Functional	Performs the entire set of tests offline.

graphicstest Command-Line Syntax

/opt/SUNWvts/bin/sparcv9/graphicstest *standard_arguments* **-o dev=***device_name* **open=***Enable/Disable* **dma=***Enable/Disable* **mem=***Enable/Disable* **chip=***Enable/Disable* **B=***n* **F=***n* **S=***value*

TABLE 22-3 graphicstest 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.
open=*Enable/Disable*	Enables or disables the open test. The default value is Enable.
dma=*Enable/Disable*	Enables or disables the dma test. The default value is Enable.
mem=*Enable/Disable*	Enables or disables the memory test. The default value is Enable.
chip=*Enable/Disable*	Enables or disables the chip test. The default is Enable.
B=*n*	Defines the number of times to repeat each test loop. The default value is 1.
F=*n*	Defines the number of times to repeat each subtest. The default is one.
S=*value*	Sets a bit mask to select which subtests will be used. • Bit 1 – test open • Bit 2 – test dma • Bit 3 – test mem • Bit 4 – test chip For example -o S=3 would run the open and dma tests only. The default is to run all tests.

CHAPTER 23

Host to System Controller Loopback Test (`hsclbtest`)

`hsclbtest` verifies the proper functioning of the Host to System Controller interface in Sun high-performance UltraSPARC T1 (Chip-Multithreaded [CMT] Multicore Processor) CPU based entry level servers.

`hsclbtest` exercises the Host to System controller (SC) interface for making sure that the Host to SC communication path is functional and reliable. The test essentially sends and receives packets from Host to SC on SunVTS virtual channel and verifies the contents of the packets that sent and received for integrity.

`hsclbtest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

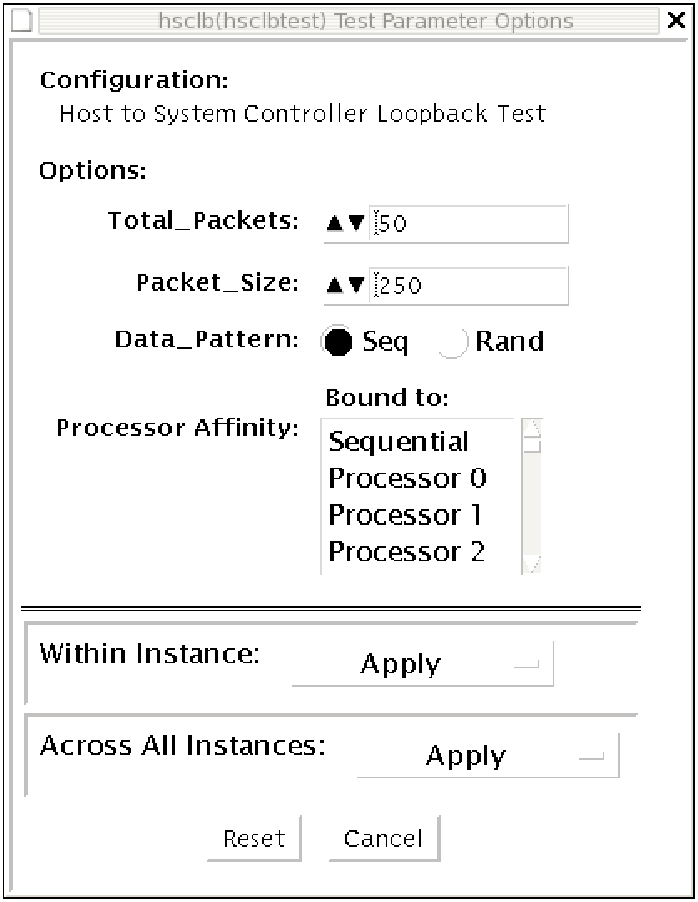


FIGURE 23-1 `hsclbtest` Test Parameter Options Dialog Box

The folloiwng table describes the `hsclbtest` test parameter options.

TABLE 23-1 `hsclbtest` Options

`hsclbtest` **Options**	**Description**
Total Packets	Specifies the total number of the packets to send. The default number of packets is 50. The maximum number of packets is 250.
Packet_Size	Specifies the size of each packet in bytes. The default packet size is 250. The minimum size is 50 and maximum size is 500.
Data_Pattern	Specifies what predefined data pattern to use for the packet contents. The options are Sequential or Random. The default is the Sequential pattern.

`hsclbtest` Test Modes

TABLE 23-2 `hsclbtest` Supported Test Modes

Test Mode	Description
Connection	The Connection test mode is supported.
Functional	Performs the full test.

`hsclbtest` Command-Line Syntax

`/opt/SUNWvts/bin/hsclbtest` *standard_arguments*
`-o tpkts`=*total_packets*, **`pksz`**=*packet_size*, **`dpat`**=*data_pattern*

TABLE 23-3 `hsclbtest` Command-Line Syntax

Arguments	Description
`tpkts`=*total_packets*	This argument is used to specify total number of packets to send. Valid range for total packets: 1 - 250. The default value is 50.
`pksz`=*packet_size*	This argument specifies size of each packet. Valid range for packet size is 50 - 500. The default value is 250.
`dpat`=*data_pattern*	This argument specifies what predefined data pattern to use for packet contents. Valid data patterns are "Seq" or "Rand" for Sequential data pattern or random data pattern respectively.

CHAPTER 24

I^2C Bus Test (i²ctest)

i2ctest Description

The i2ctest checks each I^2C bus for the status of its devices. i²ctest then determines any device faults based on the information it collects, and displays a report. i2ctest also detects and reports hung I^2C bus segments.

This test is scalable.

Note – Only the 64-bit version of this test is supported.

---

i2ctest Test Requirements

- You must have the SUNWpiclh, SUNWpiclr, SUNWpiclu, and SUNWpiclx picl packages installed correctly before running the test.
- Verify that the picld daemon is running by typing the following:

```
# ps -ef |grep picld
root 100077       1   0    Sep 11 ?       23:40 /usr/lib/picl/picld
```

If the daemon is not running, restart it by typing the following:

```
# /etc/init.d/picld start
```

`i2ctest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

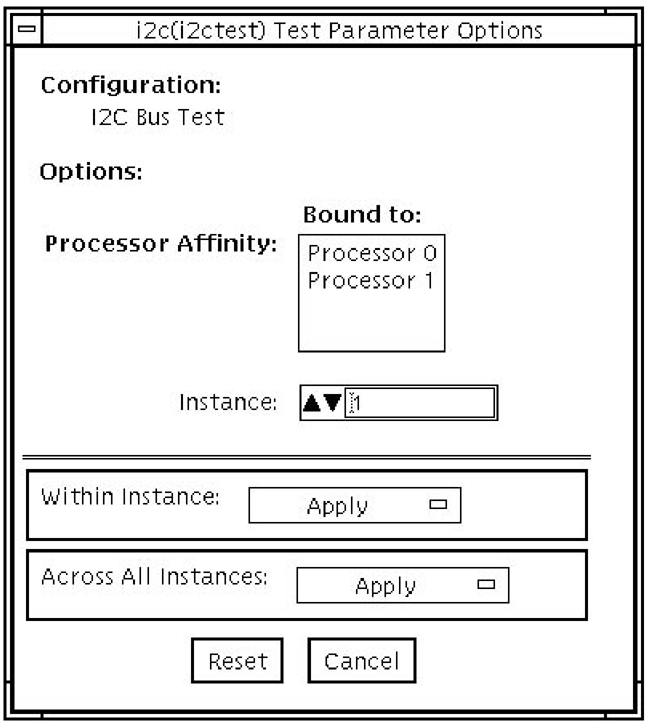


FIGURE 24-1 `i2ctest` Test Parameter Options Dialog Box

i2ctest Test Modes

TABLE 24-1 i2ctest Supported Test Modes

Test Mode	Description
Connection	Runs the full test.
Functional	Runs the full test.

i2ctest Command-Line Syntax

/opt/SUNWvts/bin/sparcv9/i2ctest *standard-arguments*

TABLE 24-2 i2ctest Command-Line Syntax

Argument	Description
standard-arguments	Specifies the standard arguments for the test [-scruvdtelnf] [-p *n*] [-i *n*] [-w *n*]. Refer to the SunVTS user's guide or use the i2ctest -u command for details.

CHAPTER 25

I^2C Inter-Integrated Circuit Test (i2c2test)

i2c2test Description

The i2c2test verifies the proper placement, operation, and data integrity on the various I^2C devices.

This test is not scalable.

i2c2test Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

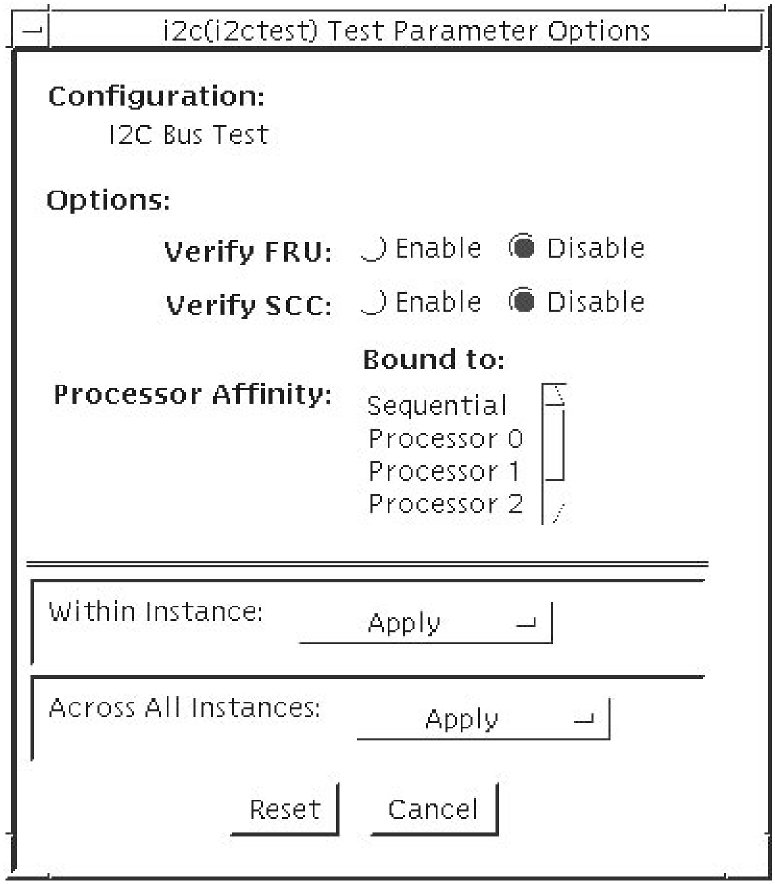


FIGURE 25-1 `i2c2test` Test Parameter Options Dialog Box

TABLE 25-1 `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 25-2 `i2c2test` Supported Test Modes

Test Mode	Description
Connection	Performs a test to verify connection to all I^2C devices.
Exclusive	Performs a test to verify connection to all I^2C 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 25-3 `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.

CHAPTER 26

Infiniband Host Channel Adapter Test (ibhcatest)

ibhcatest Description

ibhcatest comprises multiple iRISC CPU cores, two 4x Infiniband ports, and integrated SerDes components. In addition, the ibhcatest external associated components include FLASH ROM and DDR memory. ibhcatest provides high speed interconnect through PCI interface to external Infiniband fabric. Supported platforms include: two 1U and two 2U AMD Opteron entry-level servers, Sun Fire V2XX, V4XX and E-series high-end servers.

ibhcatest exercises and verifies the proper operation of the Tavor chip and its associated components, such as DDR memory, flash PROM, and internal IB packet transmit/receive circuitry. The isolates single faults to the identifiable component(s).

ibhcatest supports three execution test modes in SunVTS Connection, Exclusive and Functional. In Connection mode, the test queries for the Tavor firmware and hardware revision, and running internal loopback.

The internal loopback test is run at least once depending on the amount of time each pass takes. In Functional mode all subtests are executed according to the options selected. In Exclusive mode all subtests are executed in sequence.

Tavor supports an internal loopback mechanism that is very similar to the actual operation. The main difference is that data does not go through the integrated SerDes and the 4x IB port circuitry. On the receiving side, data does not get verified by the CRC algorithm. Otherwise, all other components of Tavor that involved in transmitting and receiving data packets are being exercised by `ibhcatest`.

Tavor-based HCA uses a single, 256 MB DDR memory for data storage at run time. This data storage is shared by three interdependent clients, Tavor driver, firmware, and hardware. During driver initialization, predetermined data structures and data are laid out in the memory.

With no exclusive access from the driver side, subsequent writes to any memory location that contain real data can cause undesirable results like a system crash. Furthermore, the data allocation size is fixed, writing to the remaining free memory does not add any value in terms of finding faults.

The memory subtest is limited to read only operations which cover the entire DDR memory. The test does not check for data corruption and no mechanism for triggering bit errors through writing to memory. The test uncovers bus related problems.

ibhcatest Subtests

TABLE 26-1 `ibhcatest` Subtests

Subtest	Description
Internal Loopback Test	The HCA supports internal loopback for packets transmitted between QPs that are assigned to the same HCA port. If a packet is being transmitted to a DLID that is equivalent to the Port LID with the LMC bits masked out or the packet DLID is a multicast LID, the packet goes on the loopback path. In this latter case, the packet also is transmitted to the fabric. When a packet is looped back, it must pass the SL2VL mapping. If the mapping yields 15 or a nonoperational VL, the packet is discarded. In the inbound direction, the ICRC and VCRC checks are blindly passed for looped back packets. Note that internal loopback is supported only for packets that are transmitted and received on the same port. Packets that are transmitted on one port and received on another port are transmitted to the fabric. The fabric directs these packets to the destination port. This subtest uses interfaces from the Tavor driver to perform loopback testing. Information such as data pattern for data packets, port number, CQ polling, retries between iteration, and the number of iterations for each `ioctl` call are passed to the driver. Once finished, status regarding the number passes completed is returned. If the number of passes does not match the number of iterations, a failure has occurred. This might happen when the number of retries is exhausted, and the last failing buffer in the retry series is returned as result. SunVTS then determines exactly what failed in the buffer and reports the failure. The options for this subtest are as follows: • `lb=`*Enabled*|*Disabled* – Turn the loopback test on or off. • `tlbport=`*1+2* – Loopback test on Port 1 and/or 2, default is 1+2. • `data=`*Pattern* – Specific data pattern (default patterns 0xa5a5a5a5). • `cq=`*Time* – Number of CQ polling time value (in microseconds per iteration). The default is 55000. The maximum is 1000000. • `loop=`*Number* – Number of loopback iterations for each pass. The default is 200. The maximum is 1000. • `warn=`*Enabled*|*Disabled* – When enabled, prints a warning message.
DDR READ Test	This subtest comprises two test modes, Sequential and Random. The start and end address offsets are determined dynamically by obtaining them from the firmware. In Sequential mode, length and starting offset are instructed by the test option `rdoffset` and `rdsz`. Then the test sequentially reads data from each memory address until all memory locations are covered or the end address is reached. Each read is accomplished by an `ioctl` call to the driver. The test returns pass or fail based on the completion status of each `ioctl` call. In Random mode, this subtest reads the number of `rdsz` times in a randomly generated address bound by the start and end address offset. The options for this subtest are as follows: • `ddr=`*Enabled*|*Disabled* – Turns the DDR memory test on or off. • `rdoffset=`*Offset* – Starts offset of DDR memory read. The default is 0x0 in hexadecimal • `rdsz=`*Size* – Read number of byte of DDR memory from Offset to Max address location. The default is 0x2000. The maximum is 256 MB in hexadecimal

Caution – In the SunVTS environment, `ibhcatest` and `nettest` are mutually exclusive. `nettest` has higher priority if the IB port interface is plumbed up when SunVTS is invoked. These two tests cannot be run at the same time at the command line If both of these tests are invoked at the command line, `ibhcatest` exits gracefully if the IB port interface is plumbed up. The commands to bring down the IB daemon (`ibd` [`IPoIB`]) are as follows:

```
# ifconfig ibdXX down
# ifconfig ibdXX unplumb
```

Where *XX* is the instance number of the interface.

`ibhcatest` Options

To reach the following dialog box, right-click on the test name in the System Map and select Test Parameter Options. Because graphics tests can test multiple types of frame buffers, the test name that is displayed corresponds to the particular frame buffer being tested. See "Frame Buffer Tests" on page 8. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system might not include the device appropriate to this test. Refer to the *SunVTS User's Guide*.

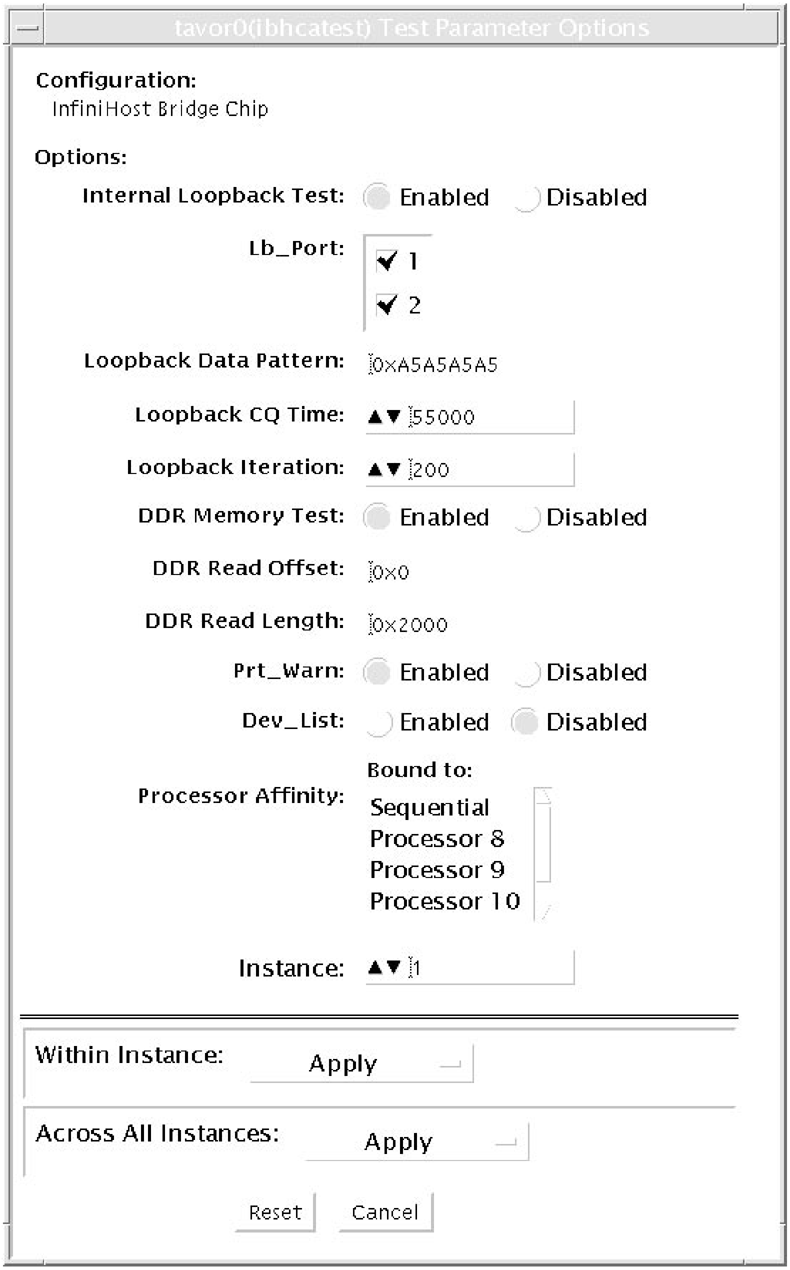


FIGURE 26-1 `ibhcatest` Test Parameter Options Dialog Box

TABLE 26-2 `ibhcatest` Options

Option	Description
`lb=`*Enabled | Disabled*	Turns the loopback test on or off.
`tlbport=`*1+2*	Loopback test on Port 1 and/or 2, default is 1+2.
`data=`*Pattern*	Specific data pattern (default patterns 0xa5a5a5a5).
`cq=`*Time*	Number of CQ polling time value (in microseconds per iteration). The default is 55000. The maximum is 1000000.
`loop=`*Number*	Number of loopback iterations for each pass. The default is 200. The maximum is 1000.
`ddr=`*Enabled | Disabled*	Turns the DDR memory test on or off.
`rdoffset=`*Offset*	Starts the offset of DDR memory read. The default is 0x0 in hexadecimal.
`rdsz=`*Size*	Read number of byte of DDR memory from Offset to Max address location. The default is 0x2000. The maximum is 256 MB in hexadecimal
`warn=`*Enabled | Disabled*	Prints a warning message when enabled.
`list`	Prints device list. No testing occurs when set.

`ibhcatest` Test Modes

TABLE 26-3 `ibhcatest` Supported Test Modes

Test Mode	Description
Connection	Queries for the Tavor firmware and hardware revision and runs the internal loopback test. The internal loopback test runs at least once depending on the duration of each pass.
Exclusive	Executes all subtests sequentially.
Functional	Executes all subtests according to what is selected.

`ibhcatest` Command-Line Syntax

ibhcatest [**-scruvdtlxnf**] [**-p** *n*] [**-i** *n*] [**-w** *n*] [**-o** [**dev=***text*] [**lb=***Enabled* | *Disabled*] [**tlbport=***1+2*] [**data=***Pattern*] [**cq=***Time*] [**loop=***Number*] [**ddr=***Enabled* | *Disabled*] [**rdoffset=***Offset*] [**rdsz=***Len*] [**warn=***Enabled* | *Disabled*] [**list**]]

Example:

```
# ibhcatest -p 0 -svf -o lb=Enabled, tlbport=1+2, data=0xA5A5A5A5,
cq=55000, loop=200, ddr=Enabled, rdoffset=0x0, rdsz=0x2000, warn=
Enabled, dev=tavor1
```

TABLE 26-4 `ibhcatest` Command-Line Syntax

Option	Description
`lb=`*Enabled* | *Disabled*	Turn the loopback test on or off.
`tlbport=`*1+2*	Loopback test on Port 1 or 2. The default is 1+2.
`data=`*Pattern*	Specific data pattern (default patterns 0xa5a5a5a5).
`cq=`*Time*	Number of CQ polling time value (in microseconds per iteration), default: 55000; max: 1000000.
`loop=`*Number*	Number of loopback iterations for each pass. The default is 200. The maximum is 1000.
`ddr=`*Enabled* | *Disabled*	Turn the DDR Memory test on or off.

TABLE 26-4 `ibhcatest` Command-Line Syntax *(Continued)*

Option	Description
`rdoffset=`*Offset*	Starts the offset of DDR memory read. Default is 0x0 in hexadecimal.
`rdsz=`*Size*	Read number of byte of DDR memory from Offset to Max address location. The default is 0x2000. The maximum is 256 MB in hexadecimal
`warn=`*Enabled* | *Disabled*	Prints a warning message when enabled.
`list`	Prints a device list. No testing occurs when set.

CHAPTER 27

Expert3D Frame Buffer Test (ifbtest)

ifbtest Description

The ifbtest verifies the functionality of the Expert3D frame buffer.

ifbtest can detect and adapt to many video modes of the Expert3D frame buffer. All tests can run at a resolution of 1024x768 or higher.

You can interrupt ifbtest 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 Expert3D accelerator port while running ifbtest. This combination causes SunVTS to return incorrect errors.

`ifbtest` 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 "Testing Multiple Frame Buffers" on page 9.

`ifbtest` Test Preparation

Perform one of the following procedures prior to performing `ifbtest` to ensure that the test runs as smoothly as possible.

▼ To Run `ifbtest` in a Window System

1. **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/ifb.conf` **file.**
2. **Ensure that no other program is running that might modify the screen during the test.**
3. **Ensure that you have permission to lock the X server.** `ifbtest` **is designed to lock the X server during testing to prevent screen changes.**

4. **The JDS login window should not be displayed during testing.**
5. **Verify that the window system is only running on one Expert3D frame buffer.**

▼ To Run `ifbtest` From the Command Line

1. **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/ifb.conf` **file.**
2. **Ensure that no other program is running that might modify the screen during the test.**
3. **Ensure that the Expert3D frame buffer being tested is not the console device. Console messages might modify the screen.**

`ifbtest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

By default, all `ifbtest` options are enabled.

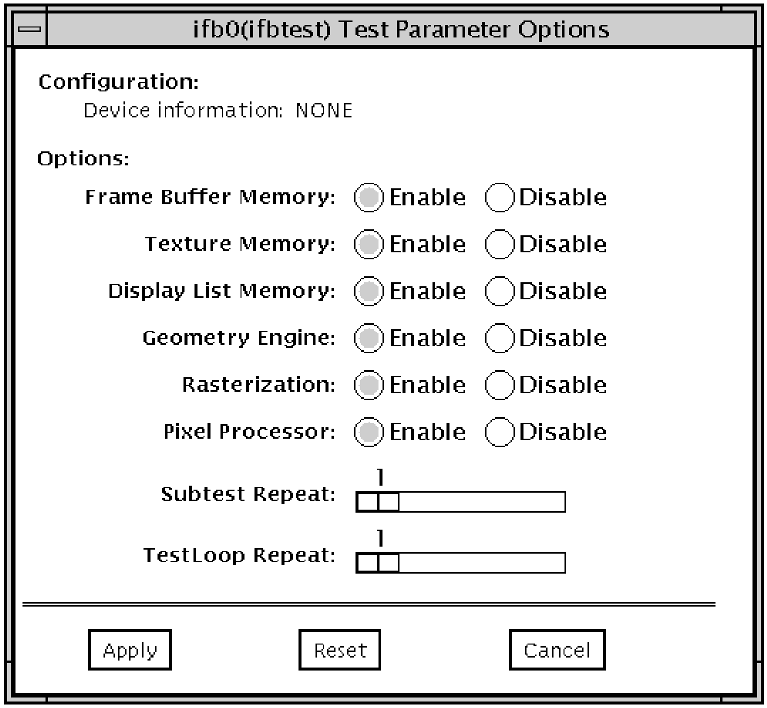


FIGURE 27-1 `ifbtest` Test Parameter Options Dialog Box

TABLE 27-1 `ifbtest` Options

`ifbtest` **Options**	**Description**
Frame Buffer Memory test	Thoroughly tests the Expert3D 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 27-1 `ifbtest` Options

`ifbtest` **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 27-1 `ifbtest` Options

`ifbtest` **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.

`ifbtest` Test Modes

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

TABLE 27-2 `ifbtest` Supported Test Modes

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

ifbtest Command-Line Syntax

/opt/SUNWvts/bin/ifbtest *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 27-3 ifbtest 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.

CHAPTER 28

Intelligent Fibre Channel Processor Test (`ifptest`)

`ifptest` Description

The `ifptest` tests the functionality of the PCI FC-AL card when there are no devices attached to the loop. The driver checks for devices on the fibre loop. If devices are detected the driver blocks any diagnostic commands.

Note – When devices are attached to the loop, do not run `ifptest`. Instead, run `disktest` tests on the individual devices. This test verifies the whole subsystem, including the FC-AL controller.

`ifptest` uses the Mailbox interface to the card. This interface enables certain firmware operations to be performed that normally would not be available to the application layer.

`ifptest` Subtests

Four subtests are run in Online and Functional modes. xxx describes the `ifptest` subtests.

TABLE 28-1 `ifptest` Subtests

Subtest	Description
Mailbox Loopback Test	Loads a series of registers into the input mailboxes on the card and then reads the output mailboxes and compares results. This test verifies that the system side of the card is operating correctly, and that the internal data paths are okay.
Firmware Revision Check	Reads the firmware revision from the firmware and compares it against a revision loaded by the driver.
ChecksuM Firmware Test	Runs an internal checksum test on the installed firmware. This test verifies that the RISC RAM on the card is fully functional and that the installed firmware is still intact. This test also serves as a quick RAM check of the RISC RAM.
Dump Revision Levels	Extracts the hardware and firmware revision levels of different submodules on the card.

`ifptest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

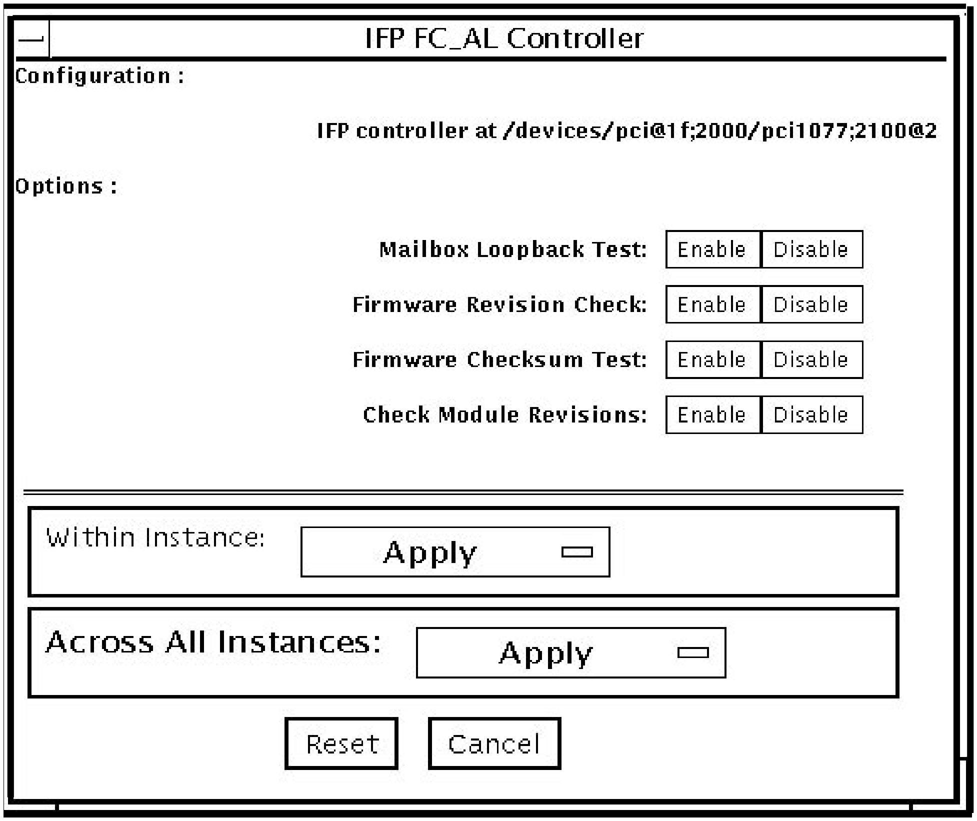


FIGURE 28-1 `ifptest` Test Parameter Options Dialog Box

TABLE 28-2 `ifptest` Options

`ifptest` **Options**	**Description**
Mailbox Loopback Test	Enables or disables the mailbox loopback command. This test writes data patterns into the mailboxes and then reads them back from the output mailboxes and verifies the data is correct.

TABLE 28-2 `ifptest` Options

`ifptest` **Options**	**Description**
FirmwaRe Revision Check	Enables or disables the firmware revision check command. This test extracts the firmware revision from the RISC firmware code and verifies against expected values.
FirmwarE Checksum Test	Enables or disables the firmware checksum command. This command instructs the interface's RISC processor to calculate the checksum on the current microcode and then compare it against the checksum that was loaded in with the microcode.
Check Module Revisions	Enables or disables the firmware check module command. This command returns the revision level of several submodules on the interface card. Although this test is executed when enabled, the module revision levels are only printed out in VERBOSE mode.

`ifptest` Test Modes

TABLE 28-3 `ifptest` Supported Test Modes

Test Mode	**Description**
Connection	Performs only an open/close operation.
Functional	Runs the full set of mailbox tests.

Note – Connection test mode only opens the controller to verify that the path is still viable.

`ifptest` Command-Line Syntax

`/opt/SUNWvts/bin/ifptest` *standard_arguments*
`-o dev=`*device name***`,mbox=`***Enable*|*Disable***`,fwrevcheck=`***Enable*|*Disable***`,`**
`checksum=`*Enable*|*Disable***`,modrevcheck=`***Enable*|*Disable*

TABLE 28-4 `ifptest` Command-Line Syntax

Argument	Description
`dev=`	The name of the device to test.
`mbox=`*Enable*|*Disable*	Enables or disables the mailbox loopback command. This test writes data patterns into the mailboxes and then reads them back from the output mailboxes and verifies the data is correct.
`fwrevcheck=` *Enable*|*Disable*	Enables or disables the firmware revision check command. This test extracts the firmware revision from the RISC firmware code and verifies against expected values.
`checksum=` *Enable*|*Disable*	Enables or disables the firmware checksum command. This command instructs the interface's RISC processor to calculate the checksum on the current microcode and then compare it against the checksum that was loaded in with the microcode.
`modrevcheck=` *Enable*|*Disable*	Enables or disables the firmware checksum command. This command returns the revision level of several sub-modules on the interface card. Although this test is executed when enabled, the module revision levels are only printed out in VERBOSE mode.

CHAPTER 29

Integer Unit Test (iutest)

iutest Description

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 following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

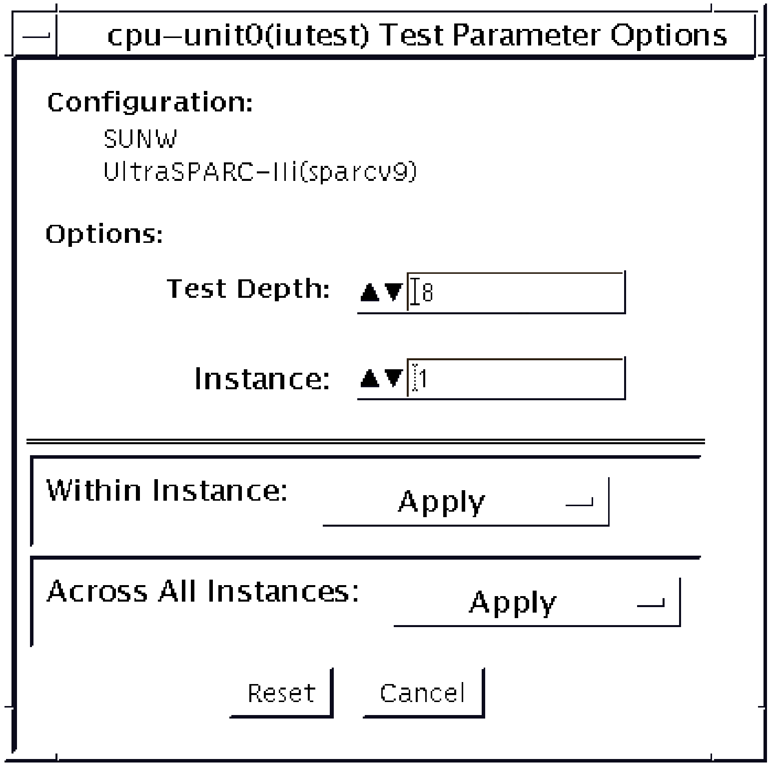


FIGURE 29-1 `iutest` Test Parameter Options Dialog Box

Test Depth is the only required test 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 29-1 `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, and so on).
Functional	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*

TABLE 29-2 describes the `iutest` command-line syntax.

TABLE 29-2 `iutest` Command-Line Syntax

Command	Description
`depth=`*val*	Specifies the value for the `depth` option. *val* is the value of the `Test_Depth` parameter option as described in the preceding `iutest` Options section.
`dev=cpu-unit`*N*	Specifies the device to test. *N* is the CPU unit number (0,1,2, and so on).

The test behavior is unpredictable if options other than those described in this section are entered.

CHAPTER 30

Sun™ XVR-1200 Graphics Accelerator Test (jfbtest)

jfbtest Description

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 "Testing Multiple Frame Buffers" on page 9.

jfbtest Test Preparation

Perform one of the following procedures to ensure that `jfbtest` runs as smoothly as possible.

▼ To Run jfbtest in a Window System

1. **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.**

2. **Verify that no other program is running that might modify the screen during the test.**

3. **Verify that you have permission to lock the X server.** `jfbtest` **is designed to lock the X server during testing to prevent screen changes.**

4. **Verify that the JDS login window should not be displayed during testing.**
5. **Verify that the window system is only running on one Sun XVR-1200 graphics accelerator.**

▼ To Run `jfbtest` From the Command Line

1. **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.**
2. **Verify that no other program is running that might modify the screen during the test.**
3. **Verify that the Sun XVR-1200 graphics accelerator being tested is not the console device. Console messages might modify the screen.**

`jfbtest` Options

To reach the following dialog box, 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 might 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.

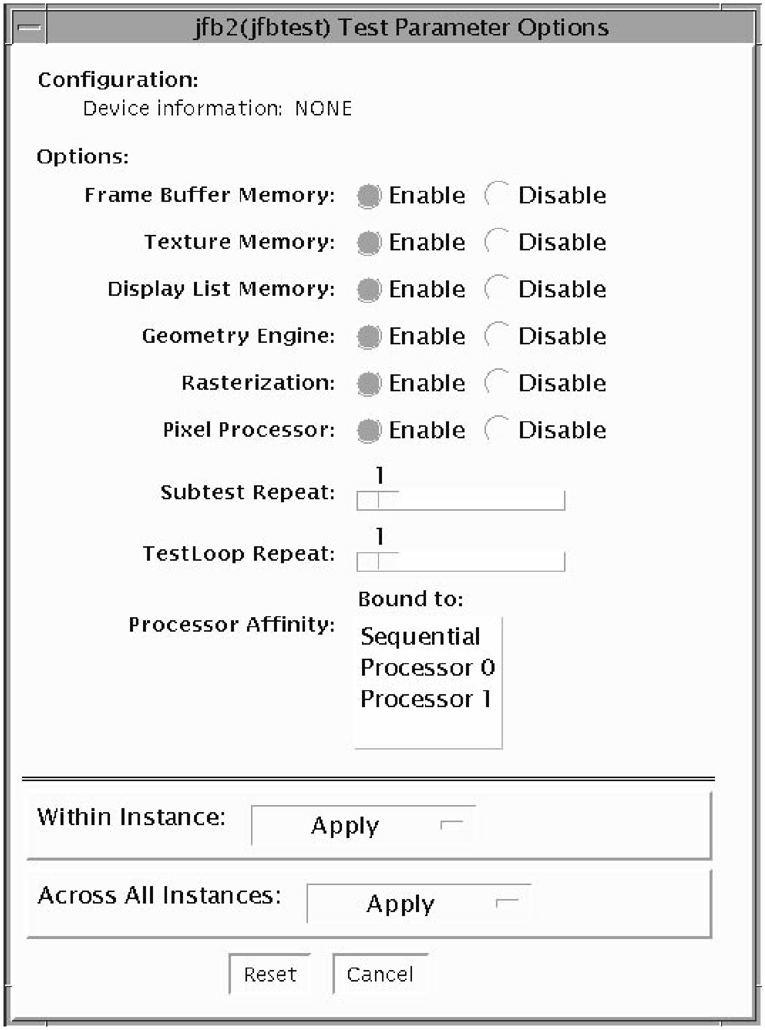


FIGURE 30-1 `jfbtest` Test Parameter Options Dialog Box

TABLE 30-1 `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. 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, 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 30-1 `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 30-1 `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 30-2 `jfbtest` Supported Test Modes

Test Mode	Description
Functional	Runs the full set of tests.
Connection	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 30-3 `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.

CHAPTER **31**

JNI 2GB FC HBA Test (`jnifctest`)

`jnifctest` Description

The `jnifctest` tests the functionality of the JNI FC HBA (JNI Fiber Channel Host Bus Adapter). There are three tests: a self-test, an internal loopback test, and an external loopback test. The self-test tests the internal consistency of the board's internal computer. The loopback tests send out blocks of data to the HBA, receives blocks echoed back to the HBA, and compares the echoed packets to the original. If `jnifctest` detects problems in the self-test, problems sending or receiving the data, or any changes in the content of the data in the loopback tests, it sends out a descriptive error message to the SunVTS console and error log.

A small list of critical patterns is most likely to detect problems on a FC network. There is also a longer list of patterns and a means inputting your data pattern for testing.

Internal loopback tests require that a loopback plug or cable be connected to the port. External loopback tests can be run on a port connected to storage, to a switch, or with a loopback plug or cable. The simplest way to get the most test coverage is to connect all ports with a cable, and run both the self-test and the external loopback test. These two tests are enabled by default.

Any storage behind a particular port is inaccessible while the tests are running. Also, system console log messages reporting renegotiation of the link status might be generated for ports connected to a switch or storage when `jnifctest` is run.

Caution – This is an exclusive mode test. This test can not be run in parallel with any other tests or applications.

`jnifctest` Subtests

`jnifctest` has three subtests described in TABLE 31-1.

TABLE 31-1 `jnifctest` Subtests

Subtest	Description
Online Selftest	Enables or disables the board self-test.
Internal Loopback Test	Enables or disables the internal loopback test.
External Loopback Test	Enables or disables the external loopback test.

`jnifctest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

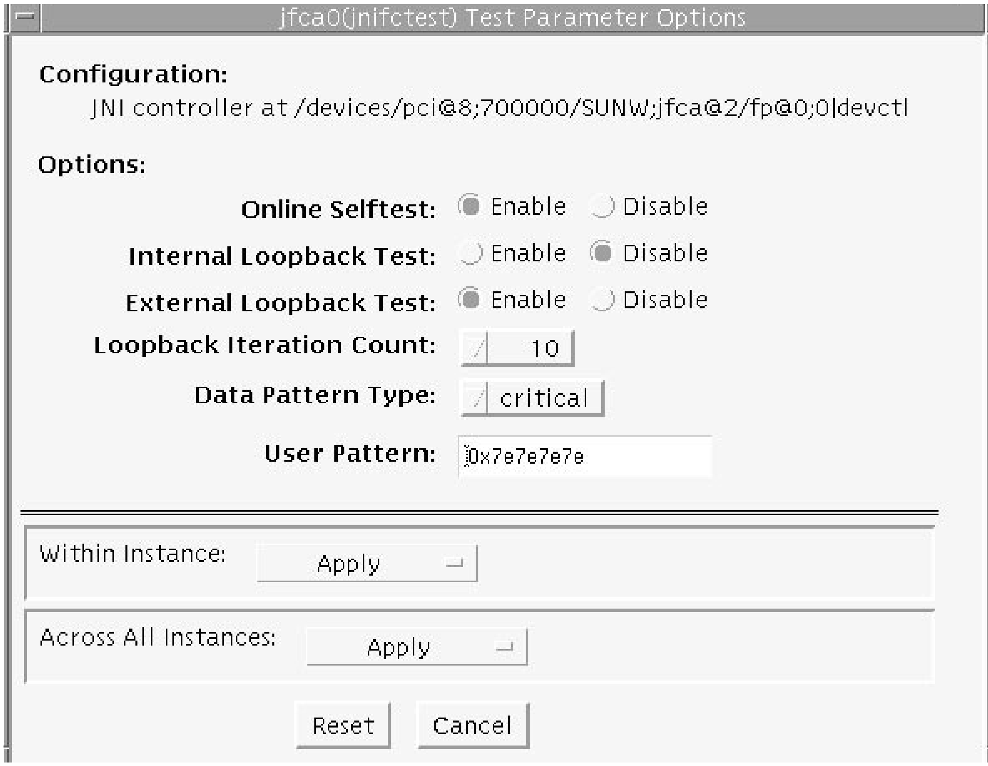


FIGURE 31-1 `jnifctest` Test Parameter Options Dialog Box

TABLE 31-2 `jnifctest` Options

`jnifctest` **Options**	**Description**
Online Selftest	Enables or disables the board self-test.
Internal Loopback Test	Enables or disables the internal loopback test.
External Loopback Test	Enables or disables the external loopback test.

TABLE 31-2 `jnifctest` Options *(Continued)*

`jnifctest` **Options**	**Description**
Loopback Iteration	Number of iterations to run the loopback test.
Data Pattern Type	What type of data to send in the loopback tests. *Critical* patterns are a dozen patterns which are optimized to cause errors in marginal FC networks. With a small number of patterns, the test will run more rapidly. *All* patterns is a list of approximately 256 patterns, including the *Critical* patterns. The *User* pattern is a single pattern that you can specify to test with.
User Pattern	A 32-bit data pattern which is used if 'user' is specified in the Data Pattern Type option. The *User Pattern* should be input in the form 0x12345678

The default values are for the Online Selftest and External Loopback test to be enabled. The Internal Loopback test is disabled. The Loopback Iteration count defaults to 10 iterations. The default Data Pattern Type is *Critical* and the default *User Pattern* is 0x7e7e7e7e.

`jnifctest` Test Modes

TABLE 31-3 `jnifctest` Supported Test Modes

Test Mode	Description
Exclusive	Runs full set of tests.

`jnifctest` Command-Line Syntax

/opt/SUNWvtshm/bin/jnifctest -vf -o dev=*jfca0*, **selftest=**{enable|disable}, **ilb=**{enable|disable}, **elb=**{enable|disable}, **iterations=**{1 - 1000000}, **selectpattern=**{critical|all|user}, **userpattern=**{*hex-value*}

TABLE 31-4 `jnifctest` Command-Line Syntax

Argument	Description
dev=device	Specifies device to be tested—for example, `jfca0`, `jfca1`, and so on.
selftest=*Enable* | *Disable*	Enables or disables the self-test.
ilb=*Enable* | *Disable*	Enables or disables the Internal loopback test.
elb=*Enable* | *Disable*	Enables or disables the External loopback test.
iterations= *1 – 1000000*	Specifies the number of iterations of the tests. The possible range of this parameter is 1 – 1,000,000. The most practical range is 10 – 5000.
selectpattern=*user* | *critical* | *all*	Specifies which data patterns are used for the loopback tests, the small list of critical hex-value data patterns, or the larger list of all hex-value data patterns. The critical hex-value data pattern list is 12 hex-value patterns. The all hex-value data pattern is a significantly larger list of hex-value data patterns.
userpattern=*hex-value*	If the `selectpattern` option is specified as *user*, this option specifies the data pattern that should be used for the loopback tests. The hex-value pattern is specified with 8 hex digits, for example, 0x12345678, 0x7e7e7e7e, or 0xcafebaba.

CHAPTER 32

Level 1 Data Cache Test (l1dcachetest)

l1dcachetest Description

l1dcachetest exercises the level 1 Data cache in the CPU module . The test writes, reads, and verifies access of multiple virtual addresses. The virtual addresses are chosen so that they cause targeted hits and misses in the cache. The test dynamically determines the size and organization of the cache and tunes the test accordingly to be effective on the l1dcache.

l1dcachetest provides data path testing of on-chip buses. With rapid move to deep sub-micron (DSM) designs, GHz clock frequencies, feature size process of 0.18 micron and below, l1dcachetest ensures the integrity of signals as they traverse conductors on a chip is becoming challenge. l1dcachetest subtests induce crosstalk noise in on-chip data buses by using Maximum Aggressor Fault (MAF) models.

l1dcahetest is self-scaling and adaptive, scaling the size of the system. l1dcachetest is multthreaded. Selection of CPU IDs is one of the options. But if that option is not specified, the test automatically retrieves the number of CPUs in the system and creates that many threads of l1dcachetest to give coverage to the whole system at a given time. The test also determines the sizes and organization of l1cache.

`l1dcachetest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

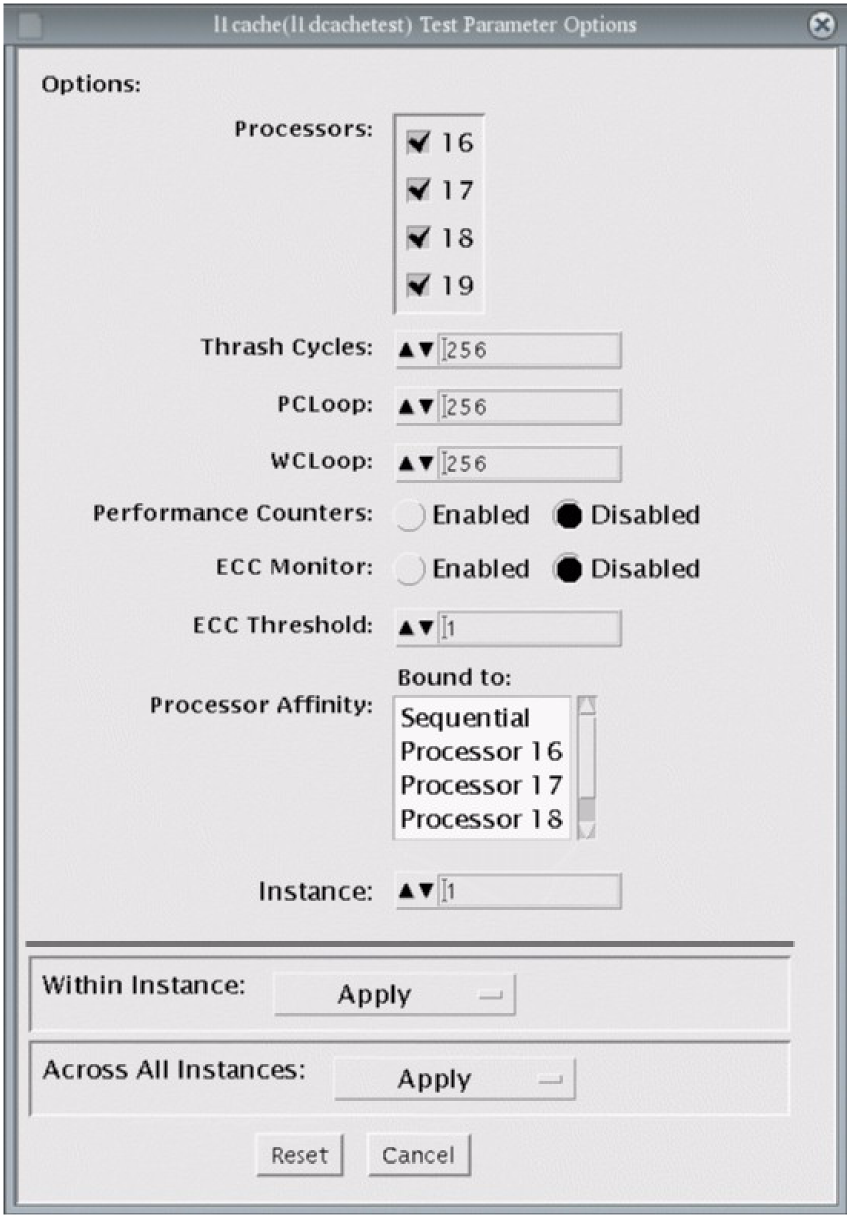


FIGURE 32-1 `l1dcachetest` Test Parameter Options Dialog Box

TABLE 32-1 `l1dcachetest` Options

Option	Description
Processors	Selects the CPU IDs for which to run this test. The test uses all CPUs on the system by default. This parameter is optional.
Thrash Cycles	Specifies the number of iterations for data cache subtests. The default is 256.
PCLoop	Specifies the number of iteration for Prefetch Cache Subtests. The default is 256. (Note – This option is displayed on system with UltraSPARC-III and UltraSPARC-IV family of processors only.
WCLoop	Specifies the number of iterations for Write Cache Subtests. The default is 256. Note: This option is displayed on systems with UltraSPARC-III and UltraSPARC-IV family of processors only.
ECC Error	Specifies whether the error messaging should be on or off. The error monitor monitors the `/var/adm/messages` file for failure messages which could be caused during test. The default is OFF.
ECC Threshold	Specifies the threshold value of the number of errors after which the test would register an error. This argument is only applicable, if the Error Monitor is on. The errors logged in the `/var/adm/messages` could be correctable errors. Ignore the errors if they are below the threshold value. The default value is 1.
Performance Counters	Enables or diables the performance counter measurements related to data cache and prefetch cache events. The default is off.

Note – Only one `l1dcachetest` gets registered for all the CPUs in the system.

Note – The `l1dcachetest` is automatically bound to a processor. Do not use the Processor Affinity option for the `l1dcachetest`.

---

l1dcachetest Test Modes

TABLE 32-2 l1dcachetest Supported Test Modes

Test Mode	Description
Connection	Performs the Connection subtest.
Exclusive	Performs only the l1dcachetest (full test).

l1dcachetest Command-Line Syntax

/opt/SUNWvts/bin/l1dcachetest *standard_arguments* [-scruvdtlxnf] [-p *n*] [-i *n*] [-w *n*] [-o **[M=***0+1+2+3+...* **],[count**=*number* **], [pcloop**=*number*], [wcloop=*number*], [**em**=*Enabled|Disabled*], [**threshold**=*1,255*], [**perf**=*Enabled|Disabled*], [**dev**=*l1cache*]]

TABLE 32-3 l1dcachetest Command-Line Syntax

Argument	Description
M=*1+2+3...*	Selects the CPU IDs for which to run this test. The test uses all CPUs on the system by default. This parameter is optional. The CPU IDs currently present in the system can be retrieved with psrinfo(1M) command. Specifying a CPU ID not present in the system or one that is offline induces an appropriate error messages from the test. For example, if you want to select CPU IDs 4, 5, 6, and 7, specify M=4+5+6+7.
count=*number*	Specifies the number of iterations for Data Cache Subtests. The default is 256.
pcloop=*number*	Specifies the number of iterations for Prefetch Cache Subtests. The default is 256. Note: This option is displayed on systems with UltraSPARC-III and UltraSPARC-IV family of processors only.
wcloop=*number*	Specifies the number of iterations for Write Cache Subtests. The default is 256. Note: This option is displayed on systems with UltraSPARC-III and UltraSPARC-IV family of processors only.
em=*Enabled|Disabled*	Specifies whether the error messaging should be on or off. The error monitor monitors the /var/adm/messages file for failure messages that could be caused during test. The default is off.

TABLE 32-3 `l1dcachetest` Command-Line Syntax

Argument	Description
threshold=*1,255*	Specifies the threshold value of the number of errors after which the test would register an error. This argument is only applicable, if the Error Monitor is on. The errors logged in the `/var/adm/messages` file could be correctable errors. Ignore the errors if they are below thrshold value. The default value is 1. If set to zero, the test still reports errors but does not stop.
perf=*Enabled* | *Disabled*	Enables or disables the Performance counter measurements related to Data Cache and Prefetch Cache events. The default is off.
dev=*l1cache*	Specifies the device to test. The default value is `l1cache`.

Note – If you do not set a value for `count` or `pcloop`, the test run with the default value of `count` and `pcloop`. To disable Data Cache subtests, specify `count=0`. For Prefetch Catch subtests, specify `pcloop=0`.

CHAPTER 33

Level 2 Cache Test (`l2sramtest`)

`l2sramtest` Description

`l2sramtest` exercises the level2 cache in the CPU module of Sun systems. In most CPUs, the level2 cache is also the external cache, but in some cases the level2 cache is on the chip. This test writes, reads, and verifies access of multiple virtual addresses. This test contains multiple subtests that try to exercise the l2cache by causing hits or misses, performing marching patterns on the l2cache cells, and writing patterns that cause electrical stress.

`l2sramtest` is self-scaling and adaptive. It scales with the size of the system. It automatically retrieves the number of CPUs in the system and internally creates that many threads of `l2sramtest` to give coverage to the whole system at a given time. For UltraSPARC-H20 and UltraSPARC-T1 systems, Level 2 cache is shared by all cores in the system thus the test retrieves the number of valid cores in the system and creates that many threads to give coverage. This test also dynamically determines the size and organization of the l2cache. You do not have to input these values.

Caution – This is an exclusive mode test. This test can not be run in parallel with any other tests or applications.

`l2sramtest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

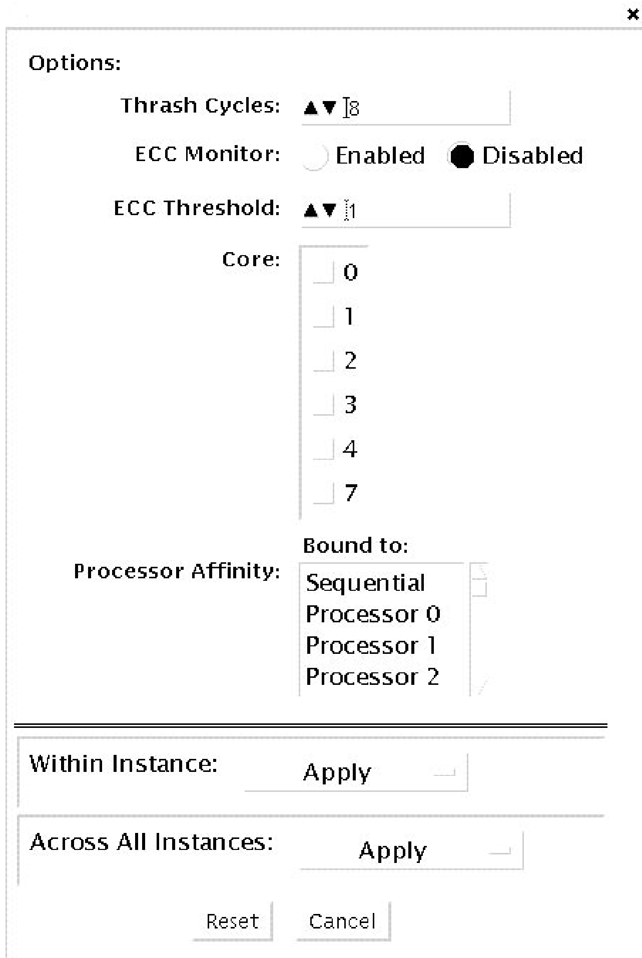


FIGURE 33-1 `l2sramtest` Test Parameter Options Dialog Box

TABLE 33-1 `l2sramtest` Options

Option	Description
Thrash Cycles	Specifies the number of thrashing cycles the test completes for the level2 cache on the system. The default value is 8.
ECC Error Monitor	Specifies whether the error monitoring should be on or off. The error monitor monitors the `/var/adm/messages` file for failure messages which could be caused due to the test. The default value is off.
Threshold	Specifies the threshold value of the number of errors after which the test would register an error. This argument is only applicable if the Error Monitor option is on. The errors that come on the `/var/adm/messages` could be correctable error, that is why the threshold value is provided for the user to give a facility to ignore the errors if they are below the threshold value. The default value is 1.
Core	This option is displayed and supported on UltraSPARC-H20 and UltraSPARC-T1 based systems only. Use the core option to perform isolation testing on selected cores only. By default, isolation testing is not enabled and system-wide testing is performed with all available cores.

Note – The `l2sramtest` automatically handles processor binding. Do not use the Processor Affinity option for the `l2sramtest`.

`l2sramtest` Test Modes

TABLE 33-2 `l2sramtest` Supported Test Modes

Test Mode	Description
Exclusive	Performs only the `l2sramtest` (full test).

`l2sramtest` Command-Line Syntax

/opt/SUNWvts/bin/sparcv9/l2sramtest -*standard_arguments* **-o** [**dev=***l2sram*, **count=***[1...1023]*, **em=***[Enabled,Disabled]*, **threshold=***[0..255]*, **core**=[*0+1+2+...*]]

Note – The core option is supported on UltraSPARC-H20 and UltraSPARC-T1 based systems only.

Note – The `l2sramtest` is not a per CPU test. There will be only one `l2sramtest` for the whole system (one image of Solaris). It will run on all the CPUs of the domain.

TABLE 33-3 `l2sramtest` Command-Line Syntax

Argument	Description
`dev=`*l2sram*	Specifies the device. The default value is `l2sram`.
`count=`*number*	Specifies the number of thrashing cycles that the test completes for the level2 cache on the system. Default value for Offline mode is 8.
`em=`*Enabled/Disabled*	Specifies the enabling or disabling of the ECC Error Monitor. The default value is Disabled.
`threshold=`*number*	Specifies the threshold value of how many correctable ECC errors can occur in the elapsed time before `l2sramtest` reports a test failure. The default value is 1.
`core`=[*0+1+2+...*]	This option is supported on UltraSPARC-H20 and UltraSPARC-T1 based systems only. Use the core option to perform isolation testing on selected cores only. By default, isolation testing is not enabled and system-wide testing is performed with all available cores.

CHAPTER 34

Level 3 Cache Test (l3sramtest)

l3sramtest Description

l3sramtest exercises the level3 cache in the CPU module of Sun's Ultra-SPARC-IV+ systems. This is an external cache, with on-chip tags.

l3sramtest runs various subtests on the cache that try to exercise the cache by causing hits/misses, performing marching patterns on the level3 cache cells and writing patterns that cause electrical stress. This test also supports Cache Interconnect Stress test using SSO patterns that targets various interconnects between level 1, level 2, and level 3 caches.

l3sramtest is self-scaling and adaptive. It scales with the size of the system. It automatically retrieves the number of CPUs in the system and internally creates that many threads to give coverage to the whole system at a given time. This test also dynamically determines the size and organization of the l3cache. You do not have to input these values.

Caution – This is an exclusive mode test. This test can not be run in parallel with any other tests or applications.

`l3sramtest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

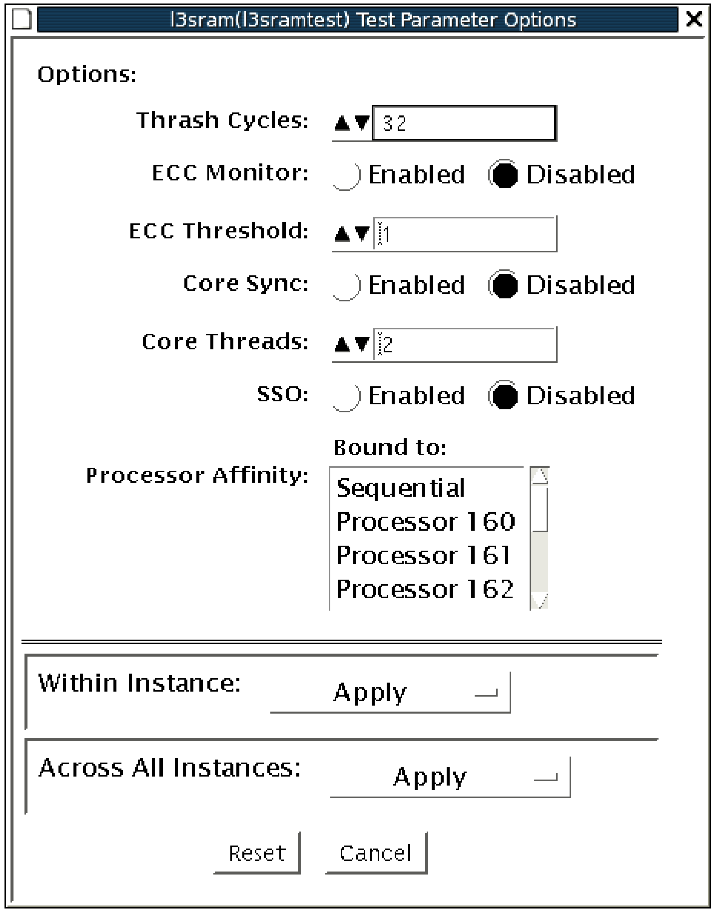


FIGURE 34-1 `l3sramtest` Test Parameter Options Dialog Box

TABLE 34-1 `l3sramtest` Options

Option	Description
Thrash Cycles	Specifies the number of thrashing cycles the test completes for the level3 cache on the system. The default value is 32.
ECC Error Monitor	Specifies whether the error monitoring should be enabled or disable. The error monitor monitors the `/var/adm/messages` file for failure messages that could be caused due to the test. The default value is disabled.
ECC Threshold	Specifies the threshold value of the number of errors after which the test would register an error. This argument is only applicable if the Error Monitor option is enabled. The errors that are logged in the `/var/adm/messages` could be correctable, that is why the threshold value is provided ignore the errors if they are below the threshold value. The default value is 1.
Core Sync	Specifies whether the core syncronization mode is enabled or disabled. When this option is enabled, each CPU core runs with exclusive access to the cache during the entire cache test execution. The default value is Disabled.
SSO	Specifies whether the Internal Cache Interconnect test is enabled or disabled. The default value is Disabled.

Note – The `l3sramtest` automatically handles processor binding. Do not use the Processor Affinity option for the `l3sramtest`.

---

l3sramtest Test Modes

TABLE 34-2 l3sramtest Supported Test Modes

Test Mode	Description
Exclusive	Performs only the l3sramtest (full test).

l3sramtest Command-Line Syntax

/opt/SUNWvts/bin/sparcv9/l3sramtest -*standard_arguments* **-o** [**dev=***l3sram*, **count=***[1...1024]*, **em=***[Enabled,Disabled]*, **threshold=***[0..255]*, **coresync**=[*0+1+2+...*]], corethreads=[*1..4*], sso=[*Enabled,Disabled*]]

Note – The l3sramtest is not a per CPU test. There will be only one l3sramtest for the whole system (one image of Solaris). l3sramtest runs on all the CPUs of the domain.

TABLE 34-3 l3sramtest Command-Line Syntax

Argument	Description
dev=*l3sram*	Specifies the device. The default value is l3sram.
count=*number*	Specifies the number of thrashing cycles that the test completes for the level3 cache on the system. Default value for offline mode is 8.
em=*Enabled/Disabled*	Specifies the enabling or disabling of the ECC Error Monitor. The default value is disabled.
threshold=*number*	Specifies the threshold value of how many correctable ECC errors can occur in the elapsed time before l3sramtest reports a test failure. The default value is 1.

TABLE 34-3 `l3sramtest` Command-Line Syntax

Argument	Description
`coresync`= [*Enabled/Disabled*]	Specifies the enabling or disabling of the core syncronization mode. When this option is enabled, each CPU core runs with exclusive access to the cache during the entire cache test execution. The default value is Disabled.
`corethreads`= *number*	Specifies the number of threads spawned per core. This is applicable in `coresync` and `sso` options only. The default value is 2.
`sso`= [*Enabled/Disabled]*	Specifies the enabling or disabling of the internal cache interconnect test. The default value is Disabled.

CHAPTER 35

LOMlite Alarm Test (`lomlitetest`)

`lomlitetest` Description

`lomlitetest` tests the proper functioning of LOMlite and LOMlite 2 system monitoring, alarms, and lights-out management (LOM) processors currently used in Netra T platforms. This test also tests the legacy TSalarms alarm card used in some Netra t 11xx systems. `lomlitetest` exercises the hardware and device drivers for the LOMlite, LOMlite 2, or TSalarms device, and tests the system and environmental monitoring functions of the device.

This test is not scalable.

Note – During offline functional testing, messages from the LOM processor are seen on the system console. This is normal and does not indicate a fault.

`lomlitetest` Requirements

The LOMlite or TSalarms device driver must be installed

In the Netra t 11xx or Netra t 14xx platforms, the LOMlite or TSalarms plug-in card must be installed

`lomlitetest` Subtests

TABLE 35-1 `lomlitetest` Subtests

Subtest	Description
Connection subtest	Run for both LOMlite and TSalarms. Attempts to open the device nodes appropriate to the device being tested.
Passive Read subtest	Attempts to read the available data from the device being tested. Device data is read as follows: All devices: Alarm state LOMlite and LOMlite 2: • Fault LED state • Power supply state • Fans state • EEPROM event log LOMlite 2 only: • Power supply voltages • Enclosure and CPU temperatures
Active Alarms subtest	Reads, inverts, and rereads each alarm state to ensure change took effect. Resets alarms to original state and checks that the state is correct. In the case of LOMlite and LOMlite 2 devices, reads the EEPROM event log and ensures that the expected events are recorded.
Active Fault LED subtest	Reads, inverts, rereads, and resets the state of the fault indicator LED. Checks that the state changes are recorded in the EEPROM event log.

lomlitetest Options

This test has no programmable options unless it is run on a multiprocessor system. On single-processor systems, the appropriate subtests are automatically selected depending upon the test mode and the type of device detected or specified on the command line.

TABLE 35-1 shows the options menu for a multi-processor system. To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

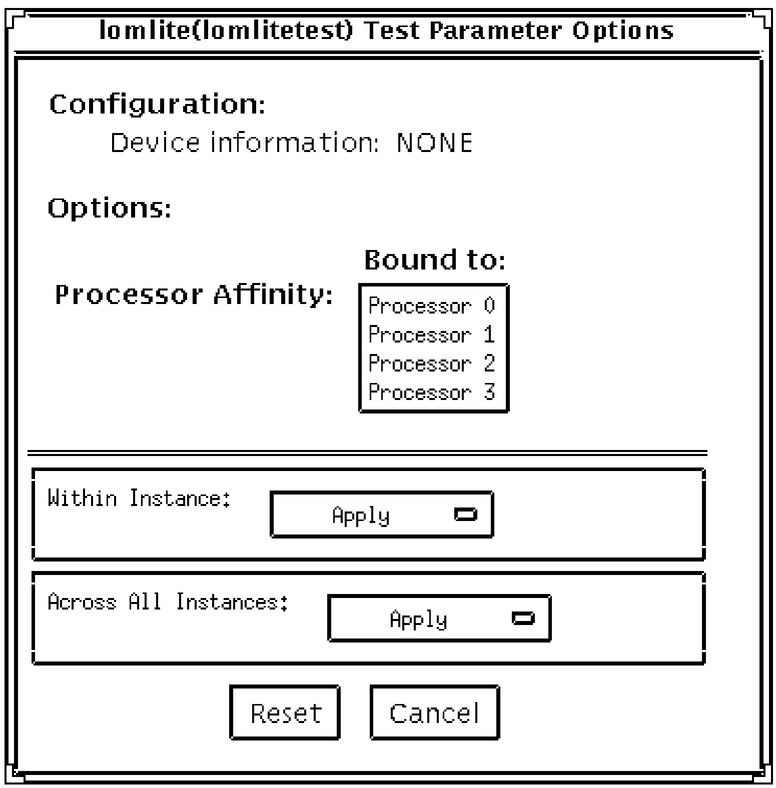


FIGURE 35-1 lomlitetest Test Parameter Options Dialog Box

`lomlitetest` Test Modes

TABLE 35-2 `lomlitetest` Supported Test Modes

Test Mode	Description
Connection	Runs the Connection subtest.
Functional	Runs all subtests.

`lomlitetest` Command-Line Syntax

`/opt/SUNWvts/bin/lomlitetest` *standard-arguments*
`-o dev=`*lomlite2* | *lomlite* | *tsalarms*

TABLE 35-3 `lomlitetest` Command-Line Syntax

Argument	Description
`dev=`*lomlite2* | *lomlite* | *tsalarms*	Selects the type of device driver to test.

CHAPTER **36**

M64 Video Board Test (`m64test`)

`m64test` Description, Requirements, and Subtests

`m64test` tests the proper functioning of the PCI-based M64 video board 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 M64 video board while running `m64test`. Do not run Power Management™ software. These programs cause 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`, add the host name to `xhost` as: **`xhost +`** *hostname.*

For full instructions on testing frame buffers, see "Testing Multiple Frame Buffers" on page 9.

`m64test` Options

To reach the following dialog box, 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 might 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.

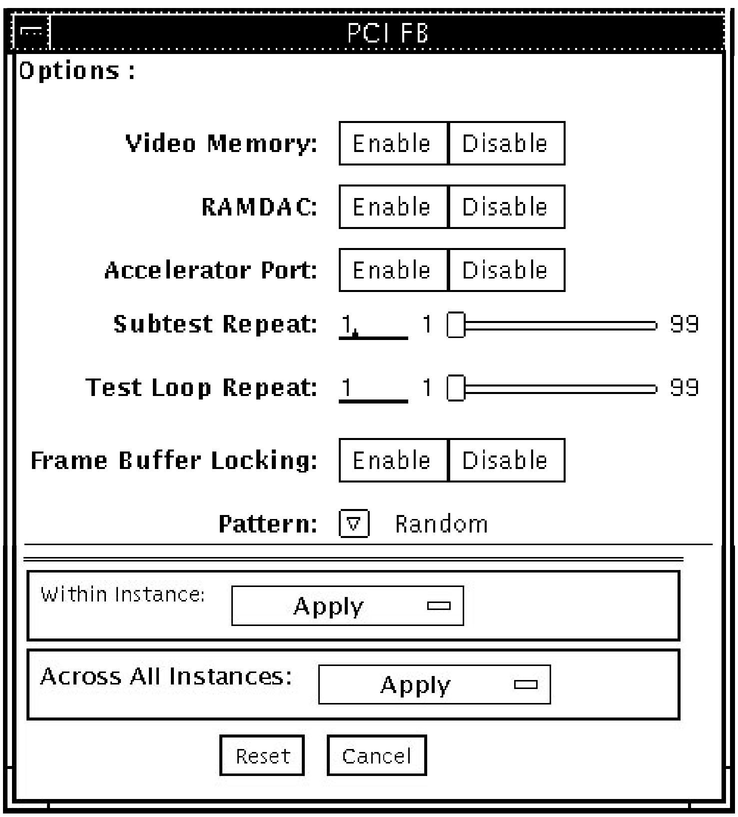


FIGURE 36-1 `m64test` Test Parameter Options Dialog Box

TABLE 36-1 `m64test` Options

`m64test` **Options**	**Description**
Video Memory test	Thoroughly tests the on-screen video memory (the memory part that is mapped on to the monitor) of the M64 video board 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. 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 three seconds. 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 300 interrupts.

TABLE 36-1 `m64test` Options

`m64test` **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) • Color comparator • 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 M64 is not the console device. When Sunvts GUI is brought up FB Locking is enabled by default if M64 is the console device. If M64 is not the console device, FB Locking is disabled by default.

`m64test` 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 Offline test mode.

TABLE 36-2 `m64test` Supported Test Modes

Test Mode	**Description**
Functional	The `m64test` verifies the M64 video board.

`m64test` Command-Line Syntax

`/opt/SUNWvts/bin/m64test` *standard-arguments* **`-o dev=`***device-name***`, S=`***subtest-number***`,F=`***subtest-loops***`,B=`***test-loops***`,L=`***disable***`,P=`***test-pattern*

TABLE 36-3 `m64test` 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 value is `m640`.
`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=0x00003` runs both test 00001 and test 00002; `n=0x00005` runs both test 0x00001 and test 0x00004. Note that you do not need the leading zeros. • `n—0x00001 VRAM` • `n—0x00002 RAMDAC` • `n—0x00004 Accelerator port test (Rendering Pipeline)` More than one test can be selected by ORing subtest numbers. For example, n = 0x00005 means VRAM and Rendering Pipeline tests. A hex number must be preceded by 0X, decimal numbers are also acceptable.
`F=`*subtest-loops*	Specifies the number of times to repeat each subtest. The default value is 1.
`B=`*test-loops*	Specifies the number of times to repeat a test loop before passing. The default is value is 1.
`L=`*disable*	Disables the frame buffer lock. Disable the lock when the m64 is not the console or when the server is not running on the m64 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 – Errors returned by `m64test` 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 M64 video board.

CHAPTER **37**

Multiprocessor Test (`mptest`)

`mptest` Description

The `mptest` verifies the proper hardware functioning of multiprocessor hardware. The test provides diagnostic test coverage for different aspects of multiprocessor functionality, such as E-cache coherency, synchronization primitives, I/O cache coherency and shared memory, and interprocessor interrupts.

The `mptest` is adaptive to different cache size and line sizes. The test causes cache coherency operations for E-Cache and I/O Cache. It also tests the synchronization primitives provided by the `sparcv8/sparcv9` architecture.

Note – `mptest` is not supported on sun4m platforms in SunVTS 5.1 PS6 or subsequent compatible versions.

Caution – This is an exclusive mode test. This test can not be run in parallel with any other tests or applications.

Caution – `mptest` by default selects the CPUXCall class of test method. If CPUXCall is selected, and `mptest` is run, the machine might seem hung for a few minutes. The duration depends on the number of CPUs.

`mptest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

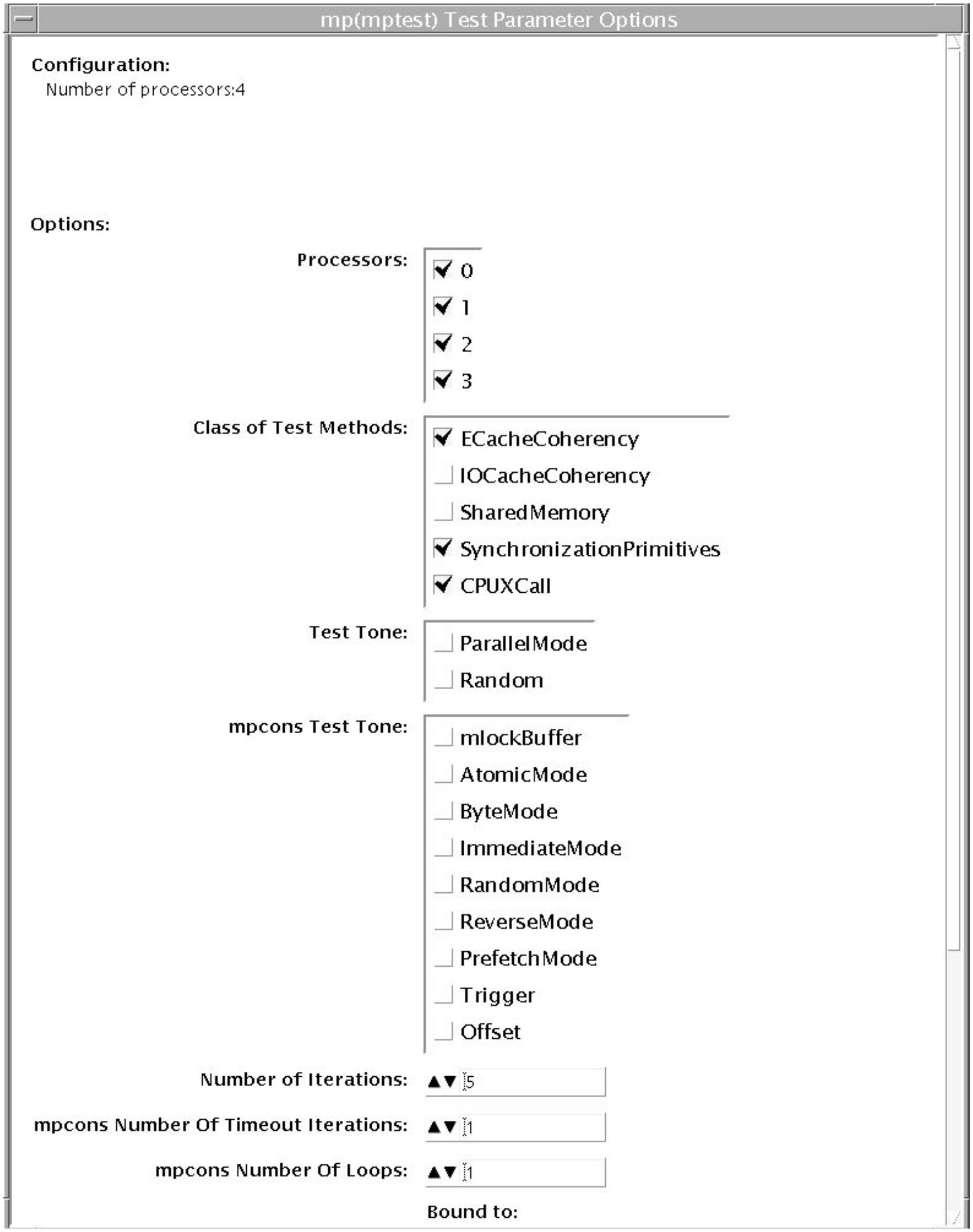


FIGURE 37-1 `mptest` Test Parameter Options Dialog Box

The processors that can be tested are listed in the Configuration area of the menu. You can enable or disable the multiprocessing test for individual processors on this menu.

The options listed in the following table can be run alone or concurrently with other options.

TABLE 37-1 `mptest` Options

`mptest` **Options**	**Description**
Processors	This option can be used to select the CPU IDs for which to run this test. The test uses all CPUs on the system by default. Thus, this parameter is optional. The CPU IDs currently present in the system can be retrieved with the `psrinfo`(1M) command. Specifying a CPU ID not present in the system or one that is currently offline induces an appropriate error message from the test.
Class of Test Methods	The Multiprocessor (MP) functionality consists of different components. A class of test method is used to specify the functionality of the MP system to be tested. Currently, the Class-of-Test methods supported by `mptest` are: E-CacheCoherency, IOCacheCoherency, SynchronizationPrimitives, and SharedMemory, and CPUXCall. This option can be used to selectively test one or more of the MP functions. If you do not specify the class of test methods, E-CacheConsistency and SynchronizationPrimitives are selected by default.
Test Tone	A test tone is a different way of executing the same test. Selecting a different test tone exercises and tests the MP functionality in a slightly different manner. The tone option can be used to select the test tone for the test. The supported test tones are Random and Parallel mode. The Random test tone introduces some randomness in testing. The ParallelTone implies that the tests perform parallel operations (like write) on different CPUs at the same time. This tone option is optional. If you do not specify any option, then the test assumes a normal tone of testing.

TABLE 37-1 `mptest` Options *(Continued)*

`mptest` **Options**	**Description**
`mpcons` Test Tone	Selects the tone for `mpconstest` cases in the `mptest`. These options are directly from the `mpconstest`. The descriptions from the `mpconstest` options are as follows: • `mlockBuffer` – Lock the shared buffer in memory • `AtomicMode` – Enable Atomic mode • `ByteMode` – Enable Byte mode • `ImmediateMode` – Enable Immediate mode • `RandomMode` – Enable Random mode • `ReverseMode` – Reverse direction to decrement through memory • `PrefetchMode` – Enable use of V9 prefetch instructions • `Trigger` – Enable LA trigger on error • `Offset` – Enable use of linesize buffer offsets This option is not mandatory. By default, the `mpcons-tone` assumes a normal mode of operation.
Number of Iterations	Same option as in `mpcontest`. Selects the number of iterations for running the test loops. The range for this option is 1 to 200 and the default is 5.
Number Of Timeout Iterations	Same option as in `mpcontest`. Selects number of timeout iterations.
CPU Wait Count 0	Same option as in `mpcontest`. Forces CPU 1 to write first if number of CPUs is less than *count*.
Number of Loops	Same option as in `mpcontest`. Sets loops to specified value.
Memory Size 0—Use Default	Same option as in `mpcontest`. Specifies memory size (MB). This should always be set to the default value.
Random Mode Seed 0	Same option as in `mpconstest`. Sets random number seed to specified value.
Strand Level Test	Performs strand level testing for multi-stranded processors such as Niagara. This option selects all strands for testing instead of selecting one strand/core. Thes option can be set to Yes or No.

mptest Test Modes

TABLE 37-2 mptest Supported Test Modes

Test Mode	Description
Exclusive	Tests the selected multiprocessor functionality.

mptest Command-Line Syntax

/opt/SUNWvts/bin/mptest *standard-arguments* M=*4+5+6+7*, method=*ECacheCoherency+IOCacheCoherency+SynchronizationPrimitives+SharedMemory+CPU Call*, tone=*ParallelMode+Random*, mpcons-tone=*mlockBuffer+AtomicMode+ByteMode+ImmediateMode+RandomMode+ReverseMode+PrefetchMode+Trigger+Offset*, count=*[1-200]*, mpcons-numtmout=*[1-10]*, mpcons-wait=*0*, mpcons-loops=*[1-999]*, mpcons-memsize=*0*, mpcons-seed=*0*, strandleveltest=*Yes*|*No*

TABLE 37-3 `mptest` Command-Line Syntax

`mptest` **Options**	**Description**
`M=`*4+5+6+7*	Selects the CPU IDs for which to run this test. The test uses all CPUs on the system by default. Hence, this parameter is optional. The CPU IDs currently present in the system can be retrieved with the `psrinfo`(1M) command. Specifying a CPU ID not present in the system or one which is currently offline induces an appropriate error message from the test. Example: If you want to select CPU IDs 4, 5, 6 and 7, specify: `M=`*4+5+6+7*. The behaviour is little different for multi-stranded processors such as Niagara. Each ID corresponds to a CPU ID of a strand in any core. If `strandleveltest` is set to No, only one CPU from a strand is chosen. To override this behavior, that is. to select more than one strand from a core, the `strandleveltest` option must be set to Yes.
`method=` *ECacheCoherency+IOCacheCoherency+SynchronizationPrimitives+SharedMemory+CPUXCall*	Selectively tests one or more of the MP functions. If you do not specify the class of test methods, E-CacheConsistency, SynchronizationPrimitives, and CPUXCall are selected by default. The Multiprocessor (MP) functionality consists of different components. A class of test method is used to specify the functionality of the MP system to be tested. The Class-of-Test methods supported by `mptest` are: E-CacheCoherency, IOCacheCoherency, SynchronizationPrimitives, SharedMemory, and CPUXCall.
`tone=`*ParallelMode+Random*	A test tone is a different way of executing the same test. Selecting a different test tone exercises and tests the MP functionality in a slightly different manner. The tone option can be used to select the test tone for the test. The supported test tones are: Random and Parallel mode. The Random test tone introduces some randomness in testing. The ParallelTone implies that the tests perform parallel operations (like write) on different CPUs at the same time. This tone option is optional. If you do not specify an option, the test assumes a normal tone of testing.

TABLE 37-3 `mptest` Command-Line Syntax *(Continued)*

`mptest` **Options**	**Description**
`mpcons-tone=` *mlockBuffer+AtomicMode+ByteMode+ImmediateMode+RandomMode+ReverseMode+PrefetchMode+Trigger+Offset*	Selects the tone for `mpconstest` cases in the `mptest`. These options are directly from the `mpconstest`. The descriptions from thes `mpconstest` options are as follows: • `mlockBuffer` – Lock the shared buffer in • memory`AtomicMode` – Enable Atomic • mode`ByteMode` – Enable Byte • mode`ImmediateMode` – Enable Immediate • mode`RandomMode` – Enable Random Mode • `ReverseMode` – Reverse direction to decrement through memory • `PrefetchMode` – Enable use of V9 prefetch instructions • `Trigger` – Enable LA trigger on error • `Offset` – Enable use of linesize buffer offsets This option is not mandatory. By default, the `mpcons-tone` assumes a normal mode of operation.
`count=`*[1-200]*	Selects the number of iterations for running the test loops. The range for this option is 1 to 200 and the default is 5.
`mpcons-numtmout=`*[1-10]*	Same option as in `mpcontest`. Selects number of timeout iterations.
`mpcons-wait=`*0*	Same option as in `mpcontest`. Forces CPU 1 to write first if number of CPUs is less than *count*.
`mpcons-loops=`*[1-999]*	Same option as in `mpcontest`. Sets loops to specified value.
`mpcons-memsize=`*0*	Same option as in `mpcontest`. Specifies memory size (MB). This should always be set to the default value.
`mpcons-seed=`*0*	Same option as in `mpcontest`. Sets random number seed to specified value.
`dev=`*mp*	Specifies the device.
`strandleveltest=`*Yes|No*	Enables the strand level testing and selects all strands for testing. The default option No performs the most optimal testing and is the most generally used. This option doesnt have any effect on non-multistranded processors such as UltraSPARC IV+.

CHAPTER 38

Sun Netra Alarm Card Test (nalmtest)

nalmtest Description

nalmtest tests the alarm card on Sun Netra 210, Sun Netra 240, and Sun Netra 440 servers.

Note – The Sun Netra Alarm Card Test (nalmtest) was called the Sun Netra 240 Alarm Card Test (n240atest) in SunVTS 5.1 PS5.

Caution – Solaris 8 2/02 OS or subsequent compatible release is required to perform the nalmtest.

nalmtest Options

To reach the following dialog box, 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. Refer to the *SunVTS User's Guide* for more details.

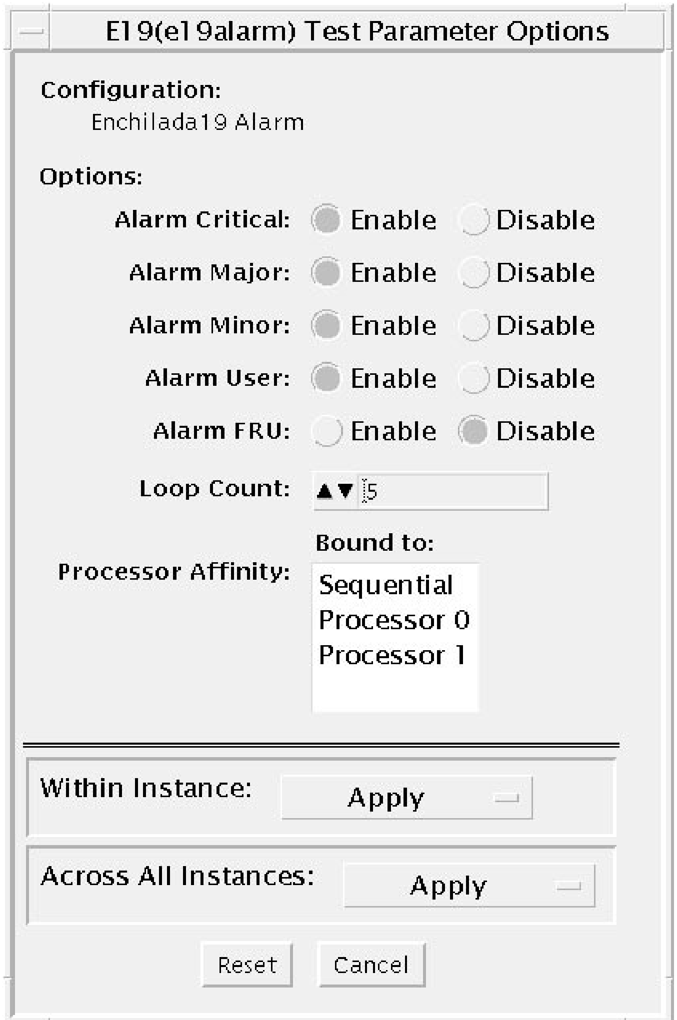


FIGURE 38-1 `nalmtest` Test Parameter Options Dialog Box

The following table describes the `nalmtest` options:

TABLE 38-1 `nalmtest` Options

`nalmtest` **Options**	**Description**
Alarm Critical	Toggles the Alarm critical LED.
Alarm Major	Toggle the Alarm critical Major.

TABLE 38-1 `nalmtest` Options *(Continued)*

`nalmtest` Options	Description
Alarm Minor	Toggles the Alarm critical Minor.
Alarm User	Toggles the Alarm critical User.
Alarm FRUID	Performs the FruID checksum test on the alarm card.
Loop Count	Sets up the loop count for toggling all four alarm LEDS. The count number is 1 to 3.

`nalmtest` Test Modes

TABLE 38-2 `nalmtest` Supported Test Modes

Test Mode	Description
Connection	Determines if the devices are connected to the system you are testing and verifies that they are accessible. Device functionality is not verified. However, you can safely run connection mode tests while the system is offline.
Functional	Tests fully exercise all aspects of the device through the associated device drivers. These tests use a significant portion of the system resources and assume that the device is available for testing. For this reason, the system must be offline with no other users or application running. This mode is sometimes referred to as Offline mode.

`nalmtest` Command-Line Syntax

/opt/SUNWvts/bin/sparcv9/nalmtest *standard_arguments* **[-o**

[dev=*device-name* **]**
[cri=*E(nable)|D(isable)* **]**
[maj=*E|D* **]**
[min=*E|D* **]**

```
[ usr=E|D ]
[ fru=E|D]
[ count=count-number] ]
```

TABLE 38-3 `nalmtest` Command-Line Syntax

Argument	Description
dev=	Specifies the name of the raw device to test.
cri=	Toggles the Alarm critical LED.
maj=	Toggles the Alarm critical Major.
min=	Toggles the Alarm critical Minor.
usr=	Toggles the Alarm critical User.
fru=	Performs FruID checksum test on the alarm card. This option does not apply to the Sun Netra 210 platform.
count=	Sets up the loop count for toggling all four alarm LEDS *count-number* is 1 to 3.

CHAPTER **39**

Ethernet Loopback Test (`netlbtest`)

`netlbtest` Description

The `netlbtest` replaces the `gemtest` previously included in SunVTS. It provides functional test coverage of the devices which have device drivers that support the Ethernet loopback test. These devices include `eri` (the Ethernet device in the RIO chip) and `ge` (Gigabit Ethernet), `ce` (GigaSwift Ethernet), `dmfe` (10/100 Mbps Ethernet), and `vca` (Sun Crypto Accelerator 4000). The `netlbtest` runs in loopback (external/internal) mode.

The `netlbtest` uses DLPI RAW mode to talk to the device driver. For the purpose of this test, a packet is defined as an Ethernet header followed by the Ethernet data payload (refer to the IEEE 802.3z standard). The test generates and sends out the desired number of packets (a tunable parameter) and expects to receive the same number of packets through the loopback interface, external or internal. If an error occurs (for example, packet mismatch or timeout), an error message indicating the type of error, its probable cause(s), and recommended action(s) is displayed on the SunVTS console.

The data sent out is generated by a random number generator, and put into a data buffer. Each time a packet is sent, it is selected from a different starting point of the data buffer, so that any two consecutively transmitted packets are not the same.

Note – Do not run `nettest` and `netlbtest` at the same time or the tests might fail.

A new debugging capability has been added in `netlbtest`. After one packet is not received, four more packets are transmitted. If all of the packets are not received within the timeout time, the test will stop with the error message, `timed out for receiving ....` If up to four packets are missing, the test will stop with an error message, `Missed %d packet(s)....` If a packet is received late and the current transmitted packet is not received, the test will stop with a warning message, `Packet delay....` If the packets arrived late but within five times the timeout value and no packet is missing, the test passes.

`netlbtest` Test Requirements

You must have the Ethernet card and the device driver installed, a loopback connector in place (external loopback only), and Intervention mode enabled before running `netlbtest`. `netlbtest` cannot run and does not appear in the GUI if the network interface is connected to a live network. `netlbtest` also requires that the Ethernet device be configured offline before running the test. Use the `ifconfig(1M)` command to bring the Ethernet device down before running `netlbtest`. Enter the following commands to bring the interface down:

```
# ifconfig interface down
# ifconfig interface unplumb
```

To run `netlbtest`, a loopback connector must be connected to the Ethernet interface when performing the external loobpback test only. A loopback connector provides the network interface driver the necessary link for testing, while maintaining isolation from a live network.

Note – The loopback connector is *not* required for both internal and external tests of the Ethernet device. The loopback connector is required for the external loopback test only.

The loopback cable for `ge` and Sun GigaSwift Ethernet MMF adapter (`ce` fiber) is based on the following specifications— multimode, duplex, 62.5/125 micron, sc connector, 850nm. The cable can be made by splitting a standard fiber optic cable in two. The two ends of the cable should be connected to the TX and RX ports of the adapter (the order does not matter), thus forming a loop.

The loopback connector for the `eri` device is a standard RJ-45 connector. See Appendix A in the *SunVTS User's Guide* for the diagram. The loopback connector for a Sun GigaSwift Ethernet UTP adapter (`ce` copper) is a standard RJ-45 with all 8 pins connected. See Appendix A of the *SunVTS User's Guide* for the diagram.

`netlbtest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

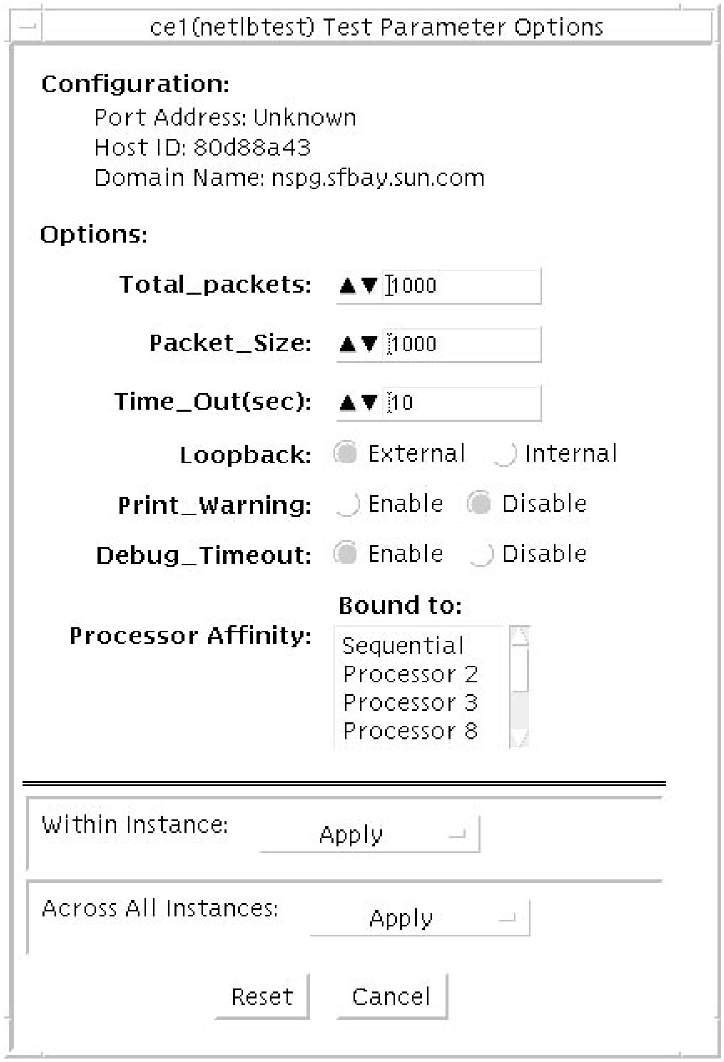


FIGURE 39-1 `netlbtest` Test Parameter Options Dialog Box

Refer to TABLE 39-1 for test parameter descriptions.

TABLE 39-1 `netlbtest` Options

`netlbtest` **Options**	**Description**
Configuration	Specifies the port address, host ID, and domain name of the system under test.
Total Packets	Specifies the total number of the packets to send. The default number of packets is 1,000. The maximum number of packets is 100,000,000.
Packet size	Determines the size (in bytes) of the packets to be transmitted. 60 <= packet size <= 1514. The default packet size is 1000 bytes.
Time_Out(sec)	Determines the amount of time (in seconds) that `netlbtest` can wait to receive packets. If no packets are received within this time frame, `netlbtest` reports an error message. The range for timeout is from 1 to 1,000 seconds.
Loopback	Determines the external and internal loopback mode. The default setting is internal loopback mode.
Print_Warning	Enables or disables the printing of warning messages. The default setting is Disable.
Processor Affinity	Binds the test to a specific processor. If no processor is specified, the test migrates between processors. This option is only available on multiprocessor systems.
Debug_Timeout	Enables or disables the debugging feature of `netlbtest`. The default setting is Disable. The range for timeout is from 1 to 1,000 seconds.

`netlbtest` Test Modes

TABLE 39-2 `netlbtest` Supported Test Modes

Test Mode	**Description**
Functional	Runs the full set of subtests. The host must not be connected to the network through the intended test device.

Since `netlbtest` requires a loopback connector for the external loopback test, it can only be selected when Intervention mode is enabled.

`netlbtest` Command-Line Syntax

`/opt/SUNWvts/bin/netlbtest` *standard-arguments*
`-o dev=`*device*, **`tpkts=`***n*, **`pksz=`***pkt-size*, **`lb=`***Internal*, **`warn=`***Disable*,
`timeout=`*number-of-seconds*

TABLE 39-3 `netlbtest` Command-Line Syntax

Argument	Description
`dev=`*device-name*	Specifies the device to test such as `ge0` or `eri0`.
`tpkts=`*n*	[1...100000], count of packets to loopback. Packets number can be 1 to 100,000. The maximum number of packets is 100,000,000.
`pksz=`*pkt-size*	Determines the size (in bytes) of the packets to be transmitted. 60 <= packet size <= 1514. The default packet size is 1000 bytes.
`lb=`*Internal*	Selects internal (or external) loopback mode.
`warn=`*Disable*	Enables or disables printing of warning messages.
`timeout=`*number-of-seconds*	Determines the amount of time (in seconds) that `netlbtest` can wait to receive packets. If no packets are received within this time frame, `netlbtest` reports an error message. The range for timeout is from 1 to 1,000 seconds.

CHAPTER 40

Network Hardware Test (nettest)

nettest Description

nettest checks all the networking hardware on the system CPU board and separate networking controllers (for example, a second SBus Ethernet controller). The machine under test must be attached to a network with at least one other system on the network.

Note – This version of nettest is used for *all* networking devices, including Ethernet (ie and le), token ring (tr, trp), quad Ethernet (QED), fiber optic (fddi, nf, bf, pf), SPARCcluster™ 1 system (em), ATM (sa, ba), HiPPI, 100-Mbits per second Ethernet (be,hme), and GigaSwift Ethernet (ce) devices.

The nettest mainly uses the Internet Control Message Protocol (ICMP), and requires at least two machines on a network—the machine under test and another machine reliable enough to be a test target. Both machines must support the Transmission Control Protocol/Internet Protocol (TCP/IP) (ICMP is part of TCP/IP). The target machine must either be configured to respond to ICMP broadcast or to RPC broadcast.

First nettest determines the target machine(s) to test against. If no targets are specified, it sends an ICMP broadcast to find them. If the test fails to find all necessary nettest targets, it tries RPC broadcast to the RPC port mapper daemon. If you specify the targets, nettest uses the specified target(s) instead.

After finding the necessary targets, `nettest` performs the following tests:

- Random test – Sends out 256 packets with random data length and random data.
- Incremental test – Sends out packets with length from minimum to maximum packet size using incremental data. (Minimum and maximum values differ for each device.)
- Pattern test – Sends 256 packets of maximum length, where each packet contains one test pattern, and all byte patterns (0 to 0xFF hex) are used. That is, the first packet contains pattern 0, the second packet contains pattern 1, and so on, until the last packet pattern of 0xFF.

Note – `nettest` is a scalable test. However, the maximum number of networked devices allowed on a system is 255, and the number of instances for each device is limited to 2. If you start the SunVTS exerciser using the `-i` option to specify a default number of instances for all tests, `nettest` cannot assign more than 2 instances per each networked device.

`nettest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

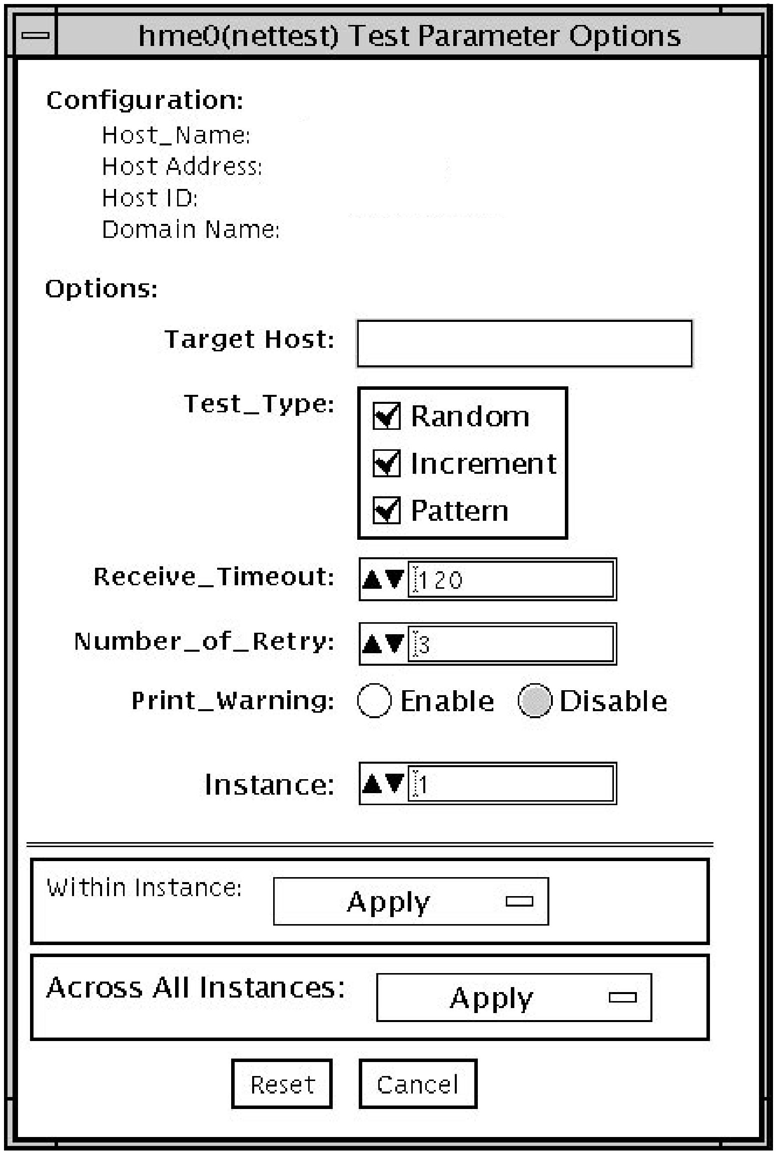


FIGURE 40-1 `nettest` Test Parameter Options Dialog Box

The Configuration section specifies the host name, host ID, host address, and domain name of the system being tested.

TABLE 40-1 `nettest` Options

`nettest` **Options**	**Description**
Target Host	Specifies one or more targets to be tested against. Target host entries can be either a host name or an Internet address. When no target host is specified, the test finds necessary targets through broadcasting. The default setting leaves this field empty.
Receive Timeout field	The default is 120 seconds, but can be changed. Use a range from 0 to 600 seconds.
Number of Retries field	The default number of retries before flagging an error is three, but can be changed. Use a range between 0 to 128 retries.
Print Warning	Disabled by default. Click Enable to see warning errors, such as retry on timeout.

`nettest` Test Modes

Both Connection and Functional test modes are supported by `nettest`. Different test schemes are performed on the network device based on the mode selected.

TABLE 40-2 `nettest` Supported Test Modes

Test Mode	**Description**
Connection	Checks whether the device is connected. It searches through all the network interfaces for a specified device name. If `nettest` does not find the device connected, the test fails; otherwise it returns: `device is connected.`
Functional	Performs all three tests (Random test, Incremental test, and Pattern test) sequentially. You may specify options that performs heavy stress testing.
Online	Performs only the Random test.

`nettest` Command-Line Syntax

`/opt/SUNWvts/bin/nettest` *standard_arguments* **`-o target=`***h1*+*h2*+*...*, **`dev=`***interface*,**`test=`***type*,**`packets=`***n*,**`pattern=`***hex*,**`timeout=`***seconds*, **`retry=`***n*,**`warn`**

TABLE 40-3 `nettest` Command-Line Syntax

Argument	Description
`target=`*h1*+*h2*+**...**	A list of test targets by host name or Internet address.
`dev=`*interface*	Network interface name. The default value is `le0` for Ethernet networks.
`test=`*type*	The test type Random, Increment, or Pattern for the desired test. The default value is Random+Increment+Pattern (all tests).
`packets=`*n*	Number of random/pattern packets. The default is 256.
`pattern=`*hex*	Specifies a data pattern, in hexadecimal form. The default is all patterns from `0` to `0xff`.
`timeout=`*seconds*	Indicates the number of seconds to wait before a timeout. The default is 120 seconds.
`retry=`*n*	Indicates the number of test timeout retries. The default is three retries.
`warn`	When enabled, prints warning messages.

CHAPTER 41

Netra CT-820 IPMI Test (nipmitest)

nipmitest Description

The nipmitest exercises and verifies the basic operation of the IPMI bus and communications between the CP2300 board to the DMC (distributed management controller) board. The test sends IPMI commands and waits for a response. The test retries up to three times no response is received.

nipmitest Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

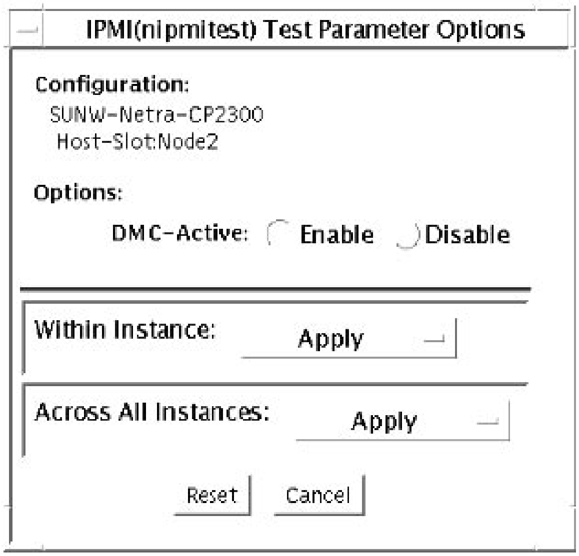


FIGURE 41-1 `nipmitest` Test Parameter Options Dialog Box

TABLE 41-1 `nipmitest` Options

`nipmitest` **Options**	**Description**
DMC Active	Enables or disables the `nipmitest`. The default is Enable.

`nipmitest` Supported Test Modes

TABLE 41-2 `nipmitest` Supported Test Modes

Test Mode	Description
Connection	Attempts to establish communication with the DMC Active Diag Daemon Actor and reports a pass or fail status.
Functional	Establishes communication with the DMC Active Diag Daemon Actor, initiates all subtests that are enabled, and reports pass or fail status.

`nipmitest` Command-Line Syntax

/opt/SUNWvtshm/bin/nipmitest -o option=*value*, **nipmitest=**{*Enable* | *Disable*}

TABLE 41-3 `nipmitest` Command-Line Syntax

Argument	Description
nipmitest=*Enable* | *Disable*	Enables or disables the `nipmitest`. The default is Enable.

CHAPTER 42

Netra Intelligent Platform Management Controller Test (nipmctest)

nipmctest Description

nipmctest exercises and verifies the proper operation of the IPMI bus and its associated components, such as sensors and FRUPROMs in the local I2C bus, and DIMM SPD information. The goal is to isolate single faults to the identifiable components.

Advanced TCA standard based on PICMG 3.x covers mechanical, power, cooling, interconnect and RASM (Reliability, Availability, Scalability, and Manageability) properties of the AdvancedTCA family specifications. ATCA Shelf management adopts the Intelligent Platform Management Interface (IPMI) 1.5 Revision 1.1 as a foundation. IPMI defines standardized and abstracted interfaces to the system management subsystem. The interfaces are the channel to monitor the health of the system hardware such as the system temperature, voltages, fans, power supplies, interconnect, system access, and recovery control.

Netra ATCA platform software provides IPMI driver and its software stack. The IPMI driver provides a user IOCTL interface—the user interface for the IPMI driver is a standard character device, nipmctest uses driver IOCTL interfaces to send IPMI commands to the IPMC and receives the responses of the commands through a system management interface connected to the IPMC.

`nipmctest` uses an in-band interface to communicate with the ShMM (Shelf Management Module) and node board in the chassis using IMPI commands. The test also uses out-of-band functions over the LAN directly to communicate with the ShMM and node boards in the chassis using IPMI commands, this feature is not yet supported on Netra ATCA products. `nipmctest` supports a common set of commands for operations such as accessing sensor values, getting thresholds, logging events, and accessing a watchdog timer.

IPMI commands supported by IPMC on Netra ATCA products are as follows:

- `netFn CMD`
- `APP(0x6)    1, 4, 8, 0x33, 0x34`
- `S/E(0x4)    1, 0x20, 0x21, 0x22, 0x2D`
- `STO(0xA)    0x10, 0x11,`
- `OEM(0x2C)   0, 1, 5, 6, 8, B, D, E, 0x12, 0x14, 0x16`

IPMI commands communicated to Active ShM and Standby ShM by `ioctl_SET_MY_ADDRESS_CMD`:

- `netFn    CMD`
- `APP(0x6)    1, 8`
- `S/E(0x4)    1`

IPMI commands encapsulated by command 0x33 and/or 0x34 (exercised by `nipmctest`):

- `netFn       CMD`
- `BRG(0x0)    0, 1`
- `APP(0x6)    1`
- `S/E(0x4)    0x2D`
- `STO(0xA)    0x20`
- `OEM(0x2C)   0x14`

`nipmctest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

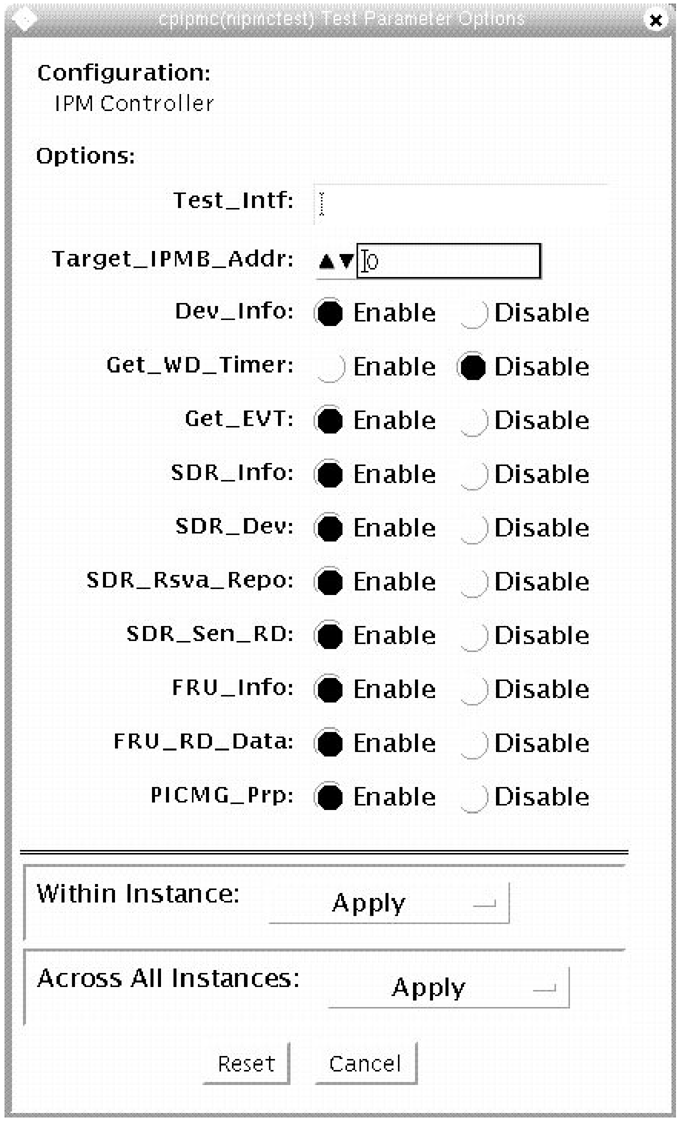


FIGURE 42-1 nipmctest Test Parameter Options Dialog Box

TABLE 42-1 `nipmctest` Options

Option	Description
Test_Intf	Specifies Open or `ipmb` interface.
Target_IPMB_Addr	Specifies the ipmb interface address of the target (response) device, rsAddr. Default: 0+0x20+0x40. Supported: 0+0x20+0x40+0x41. 0x00: accessing local IPMC by system interface 0x20: accessing ShM by ipmi command 0x34 and 0x33 through IPMB interface 0x40: accessing ShM using ioctl SET_MY_ADDRESS_CMD through IPMB interface 0x41: accessing standby ShM using ioctl SET_MY_ADDRESS_CMD through IPMB interface
Dev_Info	Enables or disables the device ID command.
Get_WD_Timer	Enables or disables the watchdog timer command. Has no effect when enabled.
Get_EVT	Enables or disables the event receiver command.
SDR_Info	Enables or disables the device SDR information.
SDR_Dev	Enables or disables the device SDR command.
SDR_Rsva_Repo	Enables or disables the device SDR repolitory command.
SDR_Sen_RD	Enables or disables the sensor reading command.
FRU_Info	Enables or disables the FRU inventory area information command.
FRU_RD_Data	Enables or disables the FRU data command.
PICMG_Prp	Enables or disables the OEM specific command (command: 0, 5, 6, 8, 0xb, 0xd, 0xe, 0x12, 0x14, 0x16).

nipmctest Test Modes

TABLE 42-2 nipmctest Supported Test Modes

Test Mode	Description
Connection	Invokes the `nipmc_bmc_get_deviceid` routine as in functional mode (`netfn:6`, `cmd:1`) and prints device and firmware information.
Functional	Performs all tests with the user specified parameters or with the default parameters.
Exclusive	Executes all subtests sequentially.

nipmctest Command-Line Syntax

/opt/SUNWvts/bin/nipmctest `-p` *1* `-scvrf -o intf=`*Open*|*ipmb*, `targ_ipmb=`*0+0x20+0x40+0x41*, `dev_info=`*Enable*|*Disable*, `wdtimer=`*Enable*|*Disable*, `get_evt=`*Enable*|*Disable*, `sdr_info=`*Enable*|*Disable*, `sdr_devs=`*Enable*|*Disable*, `sdr_rsva=`*Enable*|*Disable*, `sdr_senrd=`*Enable*|*Disable*, `fru_info=`*Enable*|*Disable*, `fru_rddata=`*Enable*|*Disable*, `ext_prp=`*Enable*|*Disable*, `dev=`*cpipmc*

TABLE 42-3 nipmctest Command-Line Syntax

Argument	Description
`intf=`*Open*|*ipmb*	Specifies Open or `ipmb` interface.
`targ_ipmb=` *0+0x20+0x40+0x41*	Specifies the ipmb interface address of the target (response) device, rsAddr. Default: 0+0x20+0x40. Supported: 0+0x20+0x40+0x41. 0x00: accessing local IPMC by system interface 0x20: accessing ShM by ipmi command 0x34 and 0x33 through IPMB interface 0x40: accessing ShM using ioctl SET_MY_ADDRESS_CMD through IPMB interface 0x41: accessing standby ShM using ioctl SET_MY_ADDRESS_CMD through IPMB interface
`dev_info=`*Enable*|*Disable*	Enables or disables the device ID command.

TABLE 42-3 `nipmctest` Command-Line Syntax

Argument	Description
`wdtimer=`*Enable|Disable*	Enables or disables the watchdog timer command. Has no effect when enabled.
`get_evt=`*Enable|Disable*	Enables or disables the event receiver command.
`sdr_info=`*Enable|Disable*	Enables or disables the device SDR information.
`sdr_devs=`*Enable|Disable*	Enables or disables the device SDR command.
`sdr_rsva=`*Enable|Disable*	Enables or disables the device SDR repolitory command.
`sdr_senrd=`*Enable|Disable*	Enables or disables the sensor reading command.
`fru_info=`*Enable|Disable*	Enables or disables the FRU inventory area information command.
`fru_rddata=`*Enable|Disable*	Enables or disables the FRU data command.
`ext_prp=`*Enable|Disable*	Enables or disables the OEM specific command (command: 0, 5, 6, 8, 0xb, 0xd, 0xe, 0x12, 0x14, 0x16).
`dev=`*cpipmc*	Psuedo name for device driver.

CHAPTER 43

PCMCIA Modem Card Test (pcsertest)

pcsertest Description

PCMCIA technology provides small, easy to use peripheral devices. The Personal Computer Memory Card International Association created this card standard for mobile computing I/O cards. These cards range from memory, FAX/modems, serial I/O, SCSI, video, sound, and so on.

The pcsertest verifies the functionality of PCMCIA modem card and PCMCIA serial I/O socket card. It does not test any other PCMCIA devices.

The pcsertest issues a series of commands to the modem to initiate a local analog loopback test and verifies this functionality.

As an option, the pcsertest tests serial I/O socket cards. This test writes a pattern of incrementing data to the serial I/O socket card, which is then looped back, read and verified.

Note – When testing serial I/O socket cards, a 9-pin loopback connector is required. However, no loopback connector is required when testing the default modem card. See Appendix A for loopback connector wiring instructions.

---

pcsertest Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

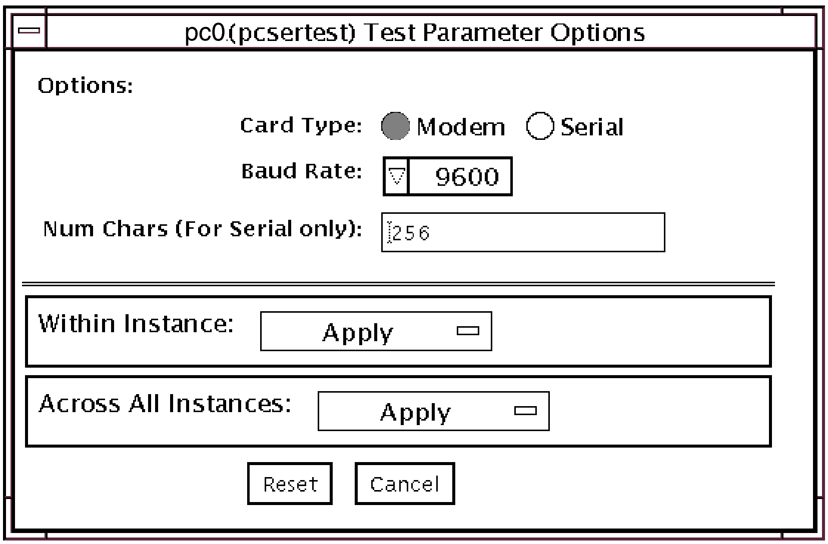


FIGURE 43-1 `pcsertest` Test Parameter Options Dialog Box

TABLE 43-1 `pcsertest` Options

`pcsertest` **Option**	**Description**
Card Type	Specifies the type of device you are testing either a PCMCIA modem, or a PCMCIA serial I/O socket card. Note – If you choose the Serial card type, and there is no serial I/O socket card in the slot, the test fails.
Baud Rate	Specifies the baud rate for testing.
Num Chars	Specifies the number of characters being used for external loopback testing of the serial socket card. By default, this is set to 256 characters. This option is applicable only to serial socket cards and ignored for the modem card.

Note – Any combination of modem and socket I/O cards can be placed in the PCMCIA slots. However, you must select the correct type of card in the Options dialog box. If you select an incorrect card type, the test fails. The default card type for each PCMCIA slot is a modem card. If only one modem card is plugged in, the empty slot is ignored.

`pcsertest` Test Mode

TABLE 43-2 `pcsertest` Supported Test Modes

Test Mode	Description
Functional	Runs the full set of tests.

`pcsertest` Command-Line Syntax

`/opt/SUNWvts/bin/pcsertest` *standard-arguments* **`-o dev=`***device-name*, **`type=`***card-type*,**`baudrate=`***speed*,**`numchars=`***n*

TABLE 43-3 `pcsertest` Command-Line Syntax

Argument	Description
`dev=`*device-name*	Specifies the device name (for example, `dev=pc0` and `dev=pc1`)
`type=`*card-type*	Specify one of the two card types for the device (`type=serial` or `type=modem`). You do not need to specify the type if the device is a modem, since modem is the default card type.
`baudrate=`*speed*	Specifies the communication speed. Specify one of the following `9600`, `19200`, `38400`, `57600`. The default is `9600`.
`numchars=`*n*	Specifies the number of characters to use for external loopback testing of the serial socket card. By default, this is set to `256` characters. This option is applicable only to serial socket cards and ignored for the modem card.

CHAPTER 44

Sun™ XVR-100 Graphics Accelerator Test (pfbtest)

pfbtest Description, Subtests, and Requirements

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.

Caution – If pfb0b is set to display, an error similar to the following occurs:
Accelerator: signature err or in test Ramdac.
Display should always be set to pfb0a when running SunVTS.

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 following dialog box, 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 might 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.

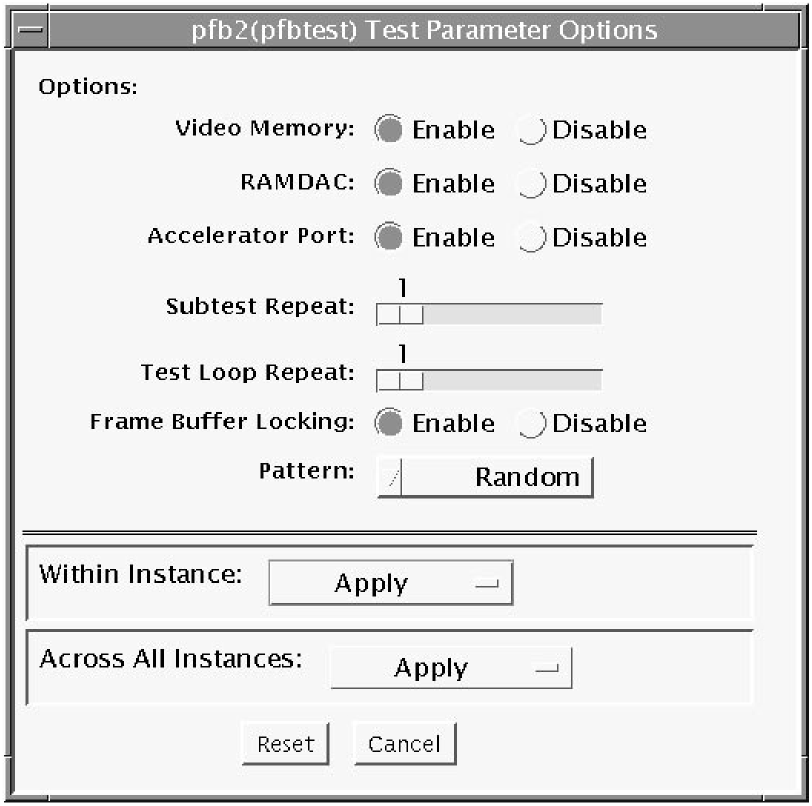


FIGURE 44-1 `pfbtest` Test Parameter Options Dialog Box

TABLE 44-1 `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).

TABLE 44-1 `pfbtest` Options *(Continued)*

`pfbtest` **Options**	**Description**
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.
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 disabled 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 44-2 `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 44-3 `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=`*subtest-loops*	Specifies the number of times to repeat each subtest. The default is 1.

TABLE 44-3 `pfbtest` Command-Line Syntax

Argument	Description
B=*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. Disables 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 – 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 45

Physical Memory Test (pmemtest)

pmemtest Description

The pmemtest checks the physical memory of the system, and reports hard and soft error correction code (ECC) errors, memory read errors, and addressing problems. The pseudo driver mem reads the physical memory.

This test reads through all the available physical memory. It does not write to any physical memory location.

pmemtest Options

pmemtest is supported both in physical mapping and logical mapping displays in the UI. In physical mapping, pmemtest provides support to test the memory on a per-board basis. Users can select the pmemtest that is displayed under the physical memory board, that is to be tested and test only that board. In logical mapping, the pmemtest options apply to the complete memory across all boards.

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

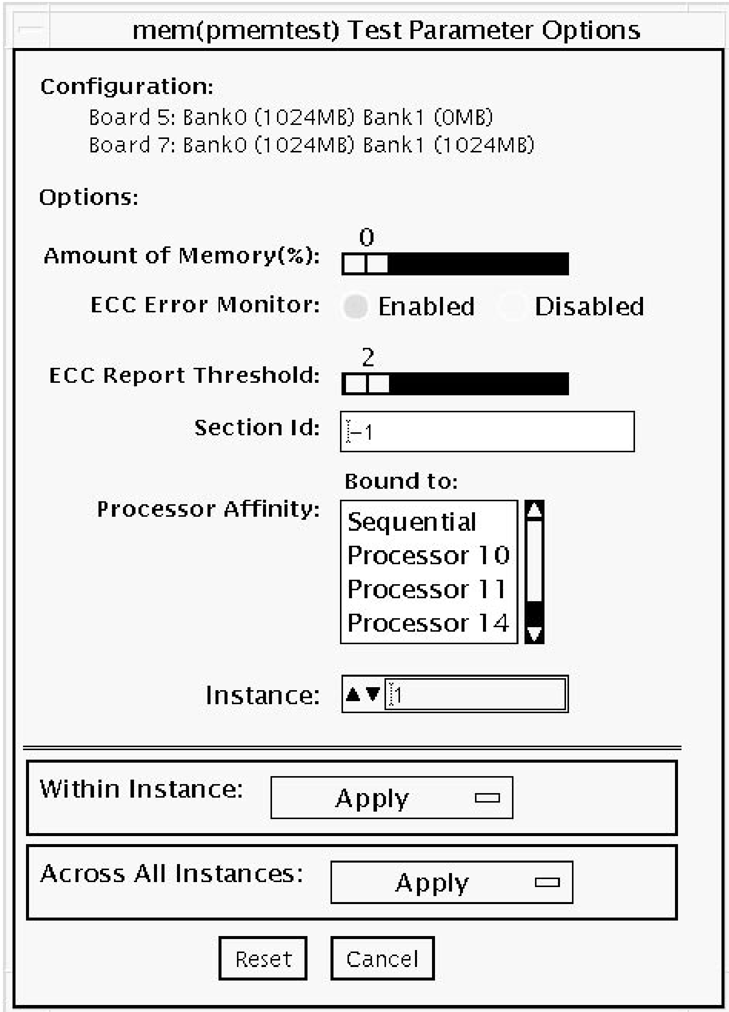


FIGURE 45-1 `pmemtest` Test Parameter Options Dialog Box

TABLE 45-1 `pmemtest` Options

`pmemtest` **Options**	**Description**
Configuration	Shows the total amount of physical memory, rounded up to the nearest MB, that is probed by the SunVTS kernel.
Amount of Memory	Specifies the percentage of the physical memory to be tested. The default 0% ensures dividing the total memory equally among instances which results in 100% coverage at the completion of every test pass. One test pass includes one pass each by all instances.
ECC Error Monitor	Enables or disables ECC error monitoring.
ECC Report Threshold	For UltraSPARC servers only. Determines how many correctable ECC errors occurred in the elapsed time before pmemtest reports a test failure. A value of zero results in no report of any correctable ECC errors. The default is 2.
Section ID	For UltraSPARC servers only. When set to -1, `pmemtest` will test one memory section in each pass, automatically testing each subsequent memory section as testing progresses. When set to a number other than -1, only the section specified is tested. A section is defined by the pass and instance number settings.
Instance	Instances are the number of copies of `pmemtest` to run simultaneously on the memory being tested.

Note – The amount of memory option is specified on a per-instance basis. The real memory coverage for one test pass depends on the amount of memory option and the number of instances. For example, if there are four instances, and each instance specifies 50% for the amount of memory option, then this will result in 200% (4 times 50%) coverage on each test pass. For guaranteed 100% memory coverage for each test pass, choose the default percentage size option as 0% for all instances.

`pmemtest` Test Modes

TABLE 45-2 `pmemtest` Supported Test Modes

Test Mode	Description
Connection	One percent of the memory is read. `pmemtest` also informs you how much physical memory is available. For sun4m, sun4u, and UltraSPARC servers, the test reports the ECC errors that have occurred since it was last invoked. The test reports ECC errors for a particular CPU or memory board when physical mapping is selected, otherwise it provides the DIMM number of the ECC memory error.
Functional	The amount of memory to be read can vary. By default 100% of the memory is tested. The test reports ECC errors for a particular memory board when physical mapping is selected, otherwise it provides the DIMM number of the ECC memory error.
Online	The amount of memory to be read can vary. By default 100% of the memory is tested. The test reports ECC errors for a particular memory board when physical mapping is selected, otherwise it provides the DIMM number of the ECC memory error.

`pmemtest` Command-Line Syntax

For 32-bit configurations:

`/opt/SUNWvts/bin/pmemtest` *standard-arguments* **`-o size=[0-100],dev=`***device-name*,**`eccmon=`***Enabled* | *Disabled***`,threshold=`***report-threshold***`,bdinfo=`***number***`,section=`***section-id*

For 64-bit configurations:

`/opt/SUNWvts/bin/sparcv9/pmemtest` *standard-arguments* **`-o size=[0-100],dev=`***device-name***`,eccmon=`***Enabled* | *Disabled***`,threshold=`***report-threshold***`,`** **`bdinfo=`***number***`,section=`***section-id*

TABLE 45-3 `pmemtest` Command-Line Syntax

Argument	Description
size=*0-100*	Specifies the percentage of memory to be tested. The default is 0% (for 100% memory coverage).
dev=*device-name*	Specifies the device to test, for example, `mem`.
eccmon=*Enabled*|*Disabled*	Enables or disables ECC error monitoring.
threshold=*report-threshold*	Number of ECC errors after which the test stops (if ECC monitor is running). When the threshold is reached the test will exit with a non zero exit code. If set to zero, the test will still report all the errors but will not stop. The default threshold is 2.
bdinfo=*number*	For UltraSPARC servers only. This argument indicates board number information. For example, if board 0 and board 5 have memory and you want the test to read the memory on both boards, then this argument should read `bdinfo=`33 (2**5+2**0). The `bdinfo` value can be specified as 0 to test the memory present on all boards.
section=*section-id*	For UltraSPARC servers only. When set to `-1`, `pmemtest` tests one memory section in each pass, automatically testing each subsequent memory section. When set to a number other than `-1`, only the section specified is tested. A section is defined by the pass and instance number settings.

CHAPTER 46

Parallel Port Printer Test (pptest)

pptest Description

The pptest exercises the parallel port devices. The test supports Sbus parallel port and IEEE 1286 complaint parallel port.

The parallel port device is an exclusive use device. Only one application can interface with it at a time.

This test is not scalable.

pptest Requirements

The SBus printer card or IEEE 1284 compliant parallel port device and device drivers must be installed to run pptest. To run the optional printer subtest, a printer must be attached to the printer port, and be powered up.

To run the optional external loopback test on IEEE 1284 compliant parallel port device, a passive loopback connector must be installed on the printer port.

Note – The external loopback test for IEEE 1284 compliant parallel port device is intended for Sun internal manufacturing use only. It requires a custom loopback connector not available to an external customer.

Large PostScript files or raster files may require that the printer has 2 MB or more of memory. Otherwise, the printout might appear on two different sheets of paper

`pptest` Subtests

`pptest` supports the following subtests.

TABLE 46-1

Subtest	Description
Internal Test FIFO Loopback	Verifies DMA and PIO accesses to the ecpp device. It uses an Internal Test FIFO on the ecpp device. There are no printer or loopback connectors required. This test is always executed on an IEEE 1284 compatible printer device.
External Passive Loopback	This verifies the parallel port I/O connections to the back panel connector. This test requires a passive loopback connector (Sun part no. 270-2965-01). This test is disabled by default and must be manually enabled by the user. The test is supported only on an IEEE 1284 compatible printer device. Note: The external loopback test is intended for Sun internal manufacturing use only. It requires a custom loopback connector that is not available to an external customer.
Printer test	This verifies the parallel port printer operation. It outputs a half page of data. The user must verify that data printed properly. This test is disabled by default and must be manually enabled by the user.

`pptest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

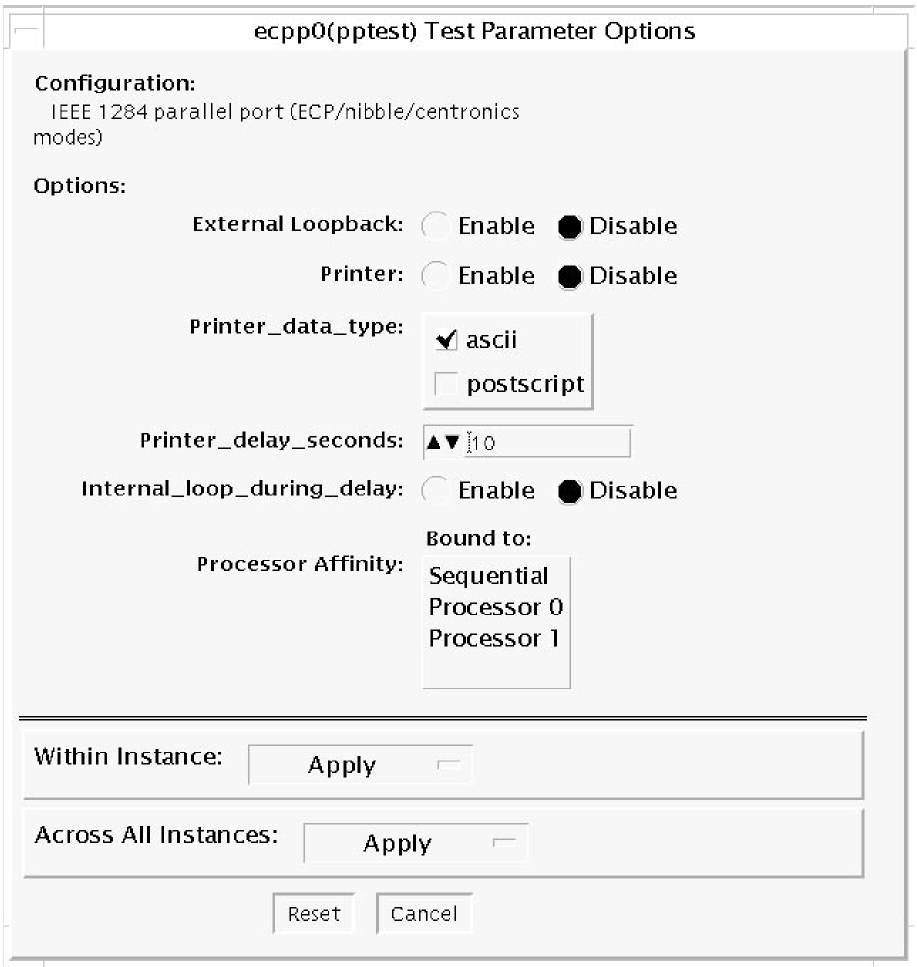


FIGURE 46-1 `pptest` Test Parameter Options Dialog Box

TABLE 46-2 `pptest` Options

Options	Description
External Loopback	Enables or disables the external loopback test. This test requires a special loopback plug. This option is available only for IEEE 1284 parallel port printer.
Printer	Enables or disables the printer test; this test requires a printer to be attached to the parallel port.
Printer_data_type	Choose whether ASCII text or PostScript data is sent to the printer. The printer test must be enabled. A postscript printer must be attached to print postscript data. This option is available only for IEEE 1284 compatible parallel printer.
Printer_delay_seconds	Selects a delay between passes of the printer test. This prevents continuous printing of data that could quickly empty the paper supply. This is only meaningful if the printer test is enabled. This option is available only for IEEE 1284 compatible parallel printer.
Interrnal_loop_during_delay	Enables the Internal Test FIFO loopback test during the printer delay time. The printer test must be enabled. This option is available only for IEEE 1284 compatible parallel printer.
Mode	Sets the print intervals. This option lets you select the intervals at which the test image is printed. • Fast – prints an image every 10 seconds. • Medium – prints an image every 12 minutes. • Extended – prints an image every 30 minutes.

pptest Test Modes

pptest test supports Connection and Functional test modes.

TABLE 46-3 Supported Test Modes

Test Mode	Description
Connection	Opens and closes the parallel port device. No data is transferred. The test passes if the device can be openned and closed successfully. The test also passes if the device cannot be opened because the device is busy with another process.
Functional	Registers a failure if the port is found busy. The unavailability of the device is interpreted as an indication of a fault condition. On ecpp(7D) devices the test runs the internal loopback test and the test can optionally run the external loopback test or the printer test.

pptest Command Line Syntax

/opt/SUNWvts/bin/sparcv9/pptest *standard_arguments* **-o dev=***dev-name*, **ext_loop=***Enable | Disable*, **printer=***Enable | Disable*, **data=***ascii*, **postscript**, **delay=***0-86400*, **dloop=***Enable | Disable*, **mode=***fast | medium | extended*

TABLE 46-4 pptest Command Line Syntax

Argument	Description
d**ev=***dev-name*	This is the target device name. The test runs on the specified device name. For Ex. dev=*ecpp0* will execute the test on /dev/ecpp0 device.
ext_loop=*Enable | Disable*	This option is available only for IEEE 1284 complaint parallel port device. If enabled, the external loopback test is run. The external loopback plug must be attached to the printer port.
printer=*Enable | Disable*	The option is available only for IEEE 1284 complaint parallel port device. If enabled, the printer test is run. A parallel port printer must be attached.
data=*ascii,postscript*	This option is available for IEEE 1284 complaint parallel port device only. Choose whether ASCII text or PostScript data is sent to the printer. A PostScript printer must be attached to print postscript data.

TABLE 46-4 `pptest` Command Line Syntax *(Continued)*

Argument	Description
delay=*0-86400*	Available for IEEE 1284 complaint parallel port devices only. Enables you to choose a delay between passes of the printer test. This delay prevents continuous printing of data that could quickly empty the paper supply.
dloop=*Enable|Disable*	Available for IEEE 1284 complaint parallel port devices only. If enabled, the printer internal loopback test is run during the print delay duration.
mode=*fast|medium|extended*	Available for non IEEE 1284 complaint parallel port devices only. Sets the test image print rate. Possible rates are as follows: • Fast – The option prints an image every 10 seconds. • Medium – The option prints an image every 12 minutes. • Extended – The option prints an image every 30 minutes.

CHAPTER 47

Qlogic Host Bus Adapter Test (qlctest)

qlctest Description

qlctest is made up of several subtests that test the functions of Qlogic ISP22xx, ISP2300, ISP2312, ISP24xx, and ISP6322 based host bus adapters. This includes all Amber and Amber2, Crystal and Crystal2, Pyramid, and Summit cards.

For qlctest to perform properly, you must have Version 4.4.5 or later of the San Foundation Kit (SFK) installed which allows HBAAPI support. To install the latest SFK, go to http://sunsolve.sun.com and download and install the software (SAN_4.4.5_install_it.tar.gz).

You must reboot the system for driver changes to take effect. Older 1 GB Fiber Channel Host Bus Adapters (FC HBAs) typically do not use the latest SFK. If you wish to test HBAs without the latest SFK installed, use the previous version of SunVTS (5.1 PS9). 2 GB FC HBAs are not displayed in qlctest without the latest SFK installed.

This test is not scalable.

Note – Do not run customer data while running qlctest, as the test takes priority over customer data requests. You cannot access data while qlctest is running.

Note – Do not run other tests while `qlctest` is running. `qlctest` might cause other tests to fail.

`qlctest` Subtests

There are nine supported subtests:

- Fcode revision check
- Firmware revision check
- Board revision check
- Checksum Firmware subtest
- Selftest
- Mailbox Loopback subtest
- Internal 10-bit Loopback subtest
- Internal 1-bit Loopback subtest
- External Loopback subtest

The Fcode, Firmware, and Board revision checks run when any other subtest is selected, and are not manually selected. The external loopback test is an intervention test. To test the Fibre loop, the QLC port must be attached to a loopback connector or to storage.

Note – A Fibre-channel loopback connector or cable is required when running internal loopback tests to prevent intermittent test failures. Refer to CR 6331949 in the release notes for additional information.

For subtest descriptions, see the `qlctest` Options table.

`qlctest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

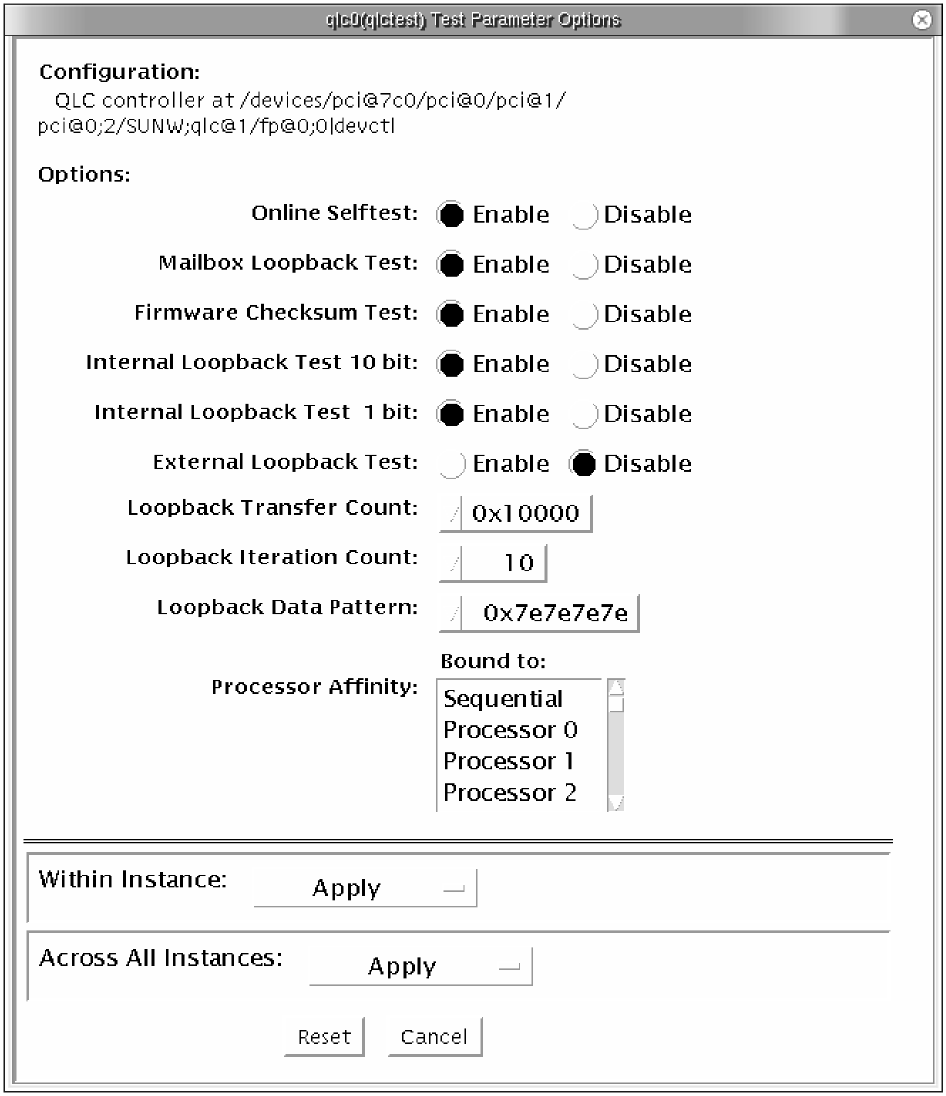


FIGURE 47-1 `qlctest` Test Parameter Options Dialog Box

TABLE 47-1 `qlctest` Options

`qlctest` **Options**	**Description**
Fcode revision check	Retrieves the fcode revision string. This core subtest is always run but not shown in the options dialog box.
Firmware revision check	Retrieves the firmware revision string. This core subtest is always run but not shown in the options dialog box.
Board revision check	Retrieves the board revision levels. This core subtest is always run but not shown in the options dialog box.
Selftest	Evaluates the functionality of ISP hardware by performing the following tests: • Transmit FIFO test • Receive FIFO test • SRAM test • Misc. Register tests Enabled by default, but can be deselected.
Mailbox Loopback subtest	Loads a series of registers into the input mailboxes on the card and then reads the output mailboxes and compares results. This verifies that the system side of the card is operating correctly, and that the internal data paths are correct. Enabled by default, but can be deselected.
Firmware Checksum subtest	Runs an internal checksum test on the installed firmware. This verifies that the RISC RAM on the card is fully functional and that the installed firmware is still intact. This test also serves as a quick RAM check of the RISC RAM. Enabled by default, but can be deselected.
Internal 10-bit Loopback subtest	Performs internal loopback test within the host adapter ISP hardware at the 10-bit interface. This test is done with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected via the test parameters menu. Enabled by default, but can be deselected.
Internal 1-bit Loopback subtest	Performs internal loopback test within the host adapter ISP hardware at the 1-bit interface. This test is done with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected via the test parameters menu. Enabled by default, but can be deselected.

TABLE 47-1 `qlctest` Options *(Continued)*

`qlctest` **Options**	**Description**
External Loopback subtest	Performs an external loopback test. This test is performed with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected via the test parameters menu. This is an intervention test, because a loopback cable is needed from the transceiver to the receiver of the QLC port when testing this port by itself. This subtest can also test the entire fibre channel loop when the loop is connected to the storage to be tested. Not enabled by default, but can be selected.
Loopback Transfer Count	Controls the packet size used in the internal 10-bit, internal 1-bit, and external loopback tests. Default value is `0x10000`.
Loopback Iteration Count	Sets the number of times to loop the internal 10-bit, internal 1-bit, and external loopback tests. Default value is `10`.
Loopback Data Pattern	Selects the data pattern to loop for the internal 10-bit, internal 1-bit, and external loopback tests. Default value is `0x7e7e7e7e`.

qlctest Test Modes

TABLE 47-2 qlctest Supported Test Modes

Test Mode	Description
Connection	Opens and closes the QLC port.
Exclusive	Supported.

qlctest Command-Line Syntax

/opt/SUNWvts/bin/qlctest *vts-system-options* **-o dev=***device-name***,selftest=***Enable*|*Disable*,**mbox=***Enable*|*Disable*,**checksum=***Enable*|*Disable*,**ilb-10=***Enable*|*Disable*,**ilb=***Enable*|*Disable*,**elb=***Enable*|*Disable*,**xcnt=***0xtransfer-count*,**icnt=***iteration-count*,**lbfpattern=***0xpattern*

TABLE 47-3 qlctest Command-Line Syntax

Argument	Description
dev	The name of the device to test.
selftest= *Enable*|*Disable*	Enables or disables the selftest command. Evaluates the functionality of the ISP hardware. Enabled by default.
mbox=*Enable*|*Disable*	Enables or disables the mailbox loopback command. This test writes data patterns into the mailboxes and then reads them back from the output mailboxes and verifies that the data is correct. Enabled by default.
checksum= *Enable*|*Disable*	Enables or disables the checksum command. Runs an internal checksum test on the installed firmware. This verifies that the RISC RAM on the card is fully functional and that the installed firmware is still intact. This test also serves as a quick RAM check of the RISC RAM. Enabled by default.
ilb-10=*Enable*|*Disable*	Enables or disables the internal 10-bit test. Performs internal loopback test within the host adapter ISP hardware at the 10-bit interface. Enabled by default.
ilb=*Enable*|*Disable*	Enables or disables the internal 1-bit test. Performs internal loopback test within the host adapter ISP hardware at the 1-bit interface. Enabled by default.

TABLE 47-3 `qlctest` Command-Line Syntax

Argument	Description
elb=*Enable | Disable*	Enables or disables the external loopback test. The desired data pattern, transfer length, and iteration count can be selected on the test parameters menu. Requires a cable for this intervention test. Disabled by default.
xcnt=*0xtransfer-count*	Controls the packet size to be transferred, for example, 0x1000. Default value is 0x10000.
icnt=*iteration-count*	Controls the number of times the loopback test will run, for example, 100. Default value is 10.
lbfpattern=*0xpattern*	Lists the data pattern to loop, for example, 0x7E7E7E7E. Default value is 0x7E7E7E7E.

CHAPTER 48

RAM Test (ramtest)

ramtest Description

ramtest is designed to stress the memory modules (RAM) instead of the whole memory subsystem. The test is optimized to achieve large memory bandwidth on UltraSPARC III (USIII) and UltraSPARC II (USII) class of CPUs. ramtest has an integrated ECC error monitor which reports the ECC errors found during the test run.

Enhanced for UltraSPARC T1 (high-end processor with chip multithreading [CMT]) based systems to use background patterns to work as the base patterns for read/write operations within the march test selected. This technique has proven to be very effective and useful for l2sram tests that use similar pattern generation techniques. You can, however, still use the previous methods.

Note – All the subtests/marches that are not intended for a particular platform are treated as an invalid option and the test provides a FATAL message indicating the invalid option.

You can now specify any march any number of times and in any order. The order of the entire command-line interface sequence is maintained for execution. This allows for a reduction in run time and provides an option for running an effective and stressful march earlier.

Caution – This is an exclusive mode test. This test can not be run in parallel with any other tests or applications.

`ramtest` Options

To reach the following dialog box, 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. Refer to the *SunVTS User's Guide* for more details.

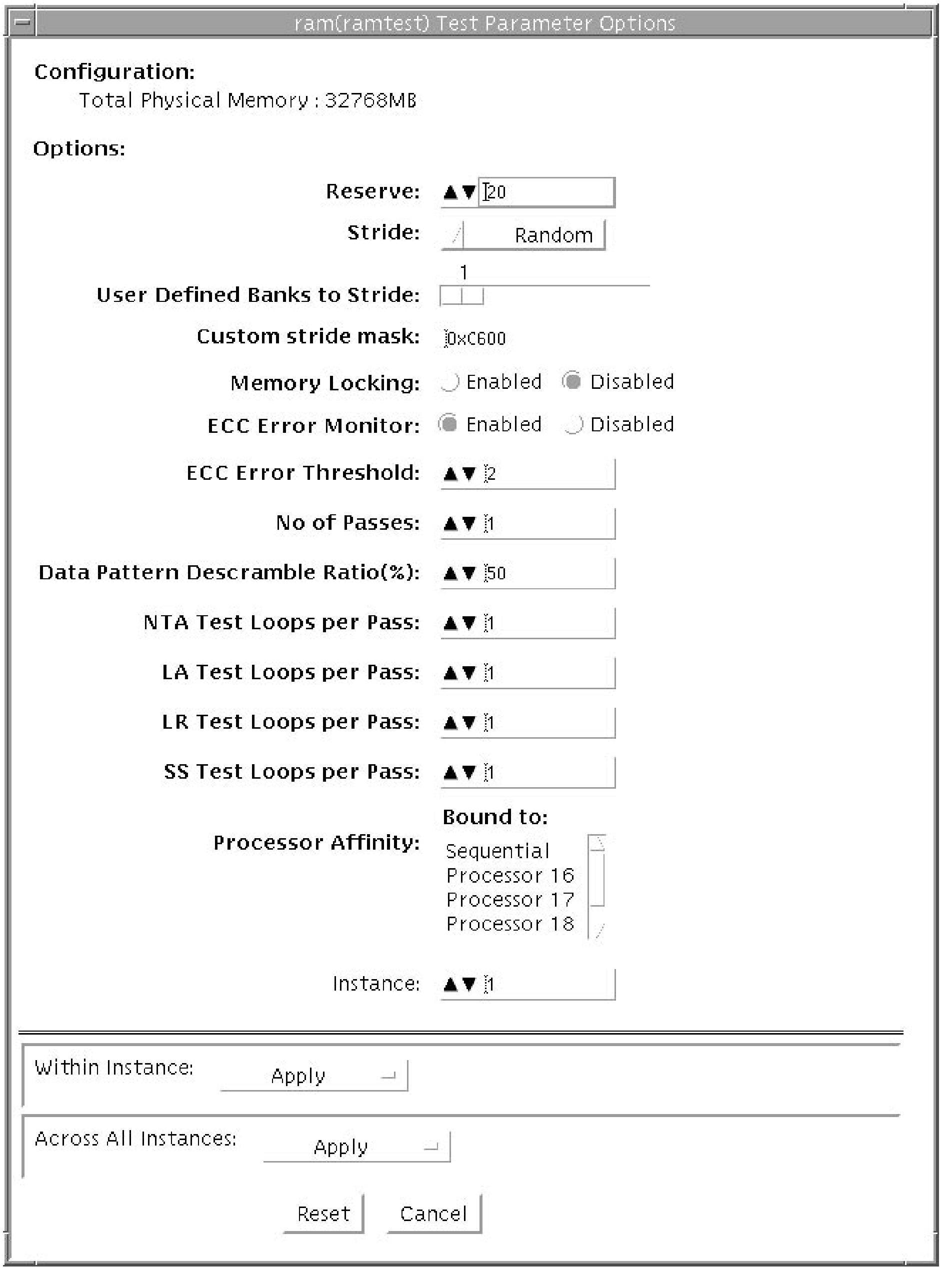


FIGURE 48-1 `ramtest` Test Parameter Options Dialog Box

The following table details the `ramtest` options:

TABLE 48-1 `ramtest` Options

`ramtest` **Options**	**Description**
Reserve	Reserve option represents the percentage of physical memory that is assumed to be in use by the OS or other processes. If you see excessive swapping while running `ramtest`, increase this percentage. The default is 20%; this means that `ramtest` allocates 80% of physical memory size for testing. Swapping decreases stress on memory and increases it on the system itself. For memory testing purposes, minimize swapping by tuning the reserve option. If the allocation or locking (in case Memory Locking is enabled) does not succeed, the amount of memory is reduced and the allocation process is repeated. Once the allocation succeeds, the amount of memory allocated is displayed in the messages.
Stride	By default this option is set to Random. It can be set to Column or Row also. In case of random, either Row or Column are randomly selected for each pass. Value of stride defines the memory locations addressed consecutively in certain subtests, in a hardware dependent manner. All testable memory is still tested. Using different strides, checks coupling among different sets of memory cells; therefore random is the recommended value for this option unless both Column and Row are being explicitly used in different instances. For FA type of uses, stride may also be set to UserDefined, in this case the test will stride the number of banks specified in the userstride option. Stride may be set to Custom in which case the stride values are randomly selected from the strides specified in the stridemask value.
User-Defined Banks to Stride	Set the number of banks that the test should stride. The value is currently limited to between 1 and 16. Row striding is not possible while using this option.

TABLE 48-1 `ramtest` Options *(Continued)*

`ramtest` **Options**	**Description**
Stride Mask	Specifies the strides used. Each thread selects one of the stride values from stridemask by selecting one of the bits in the mask. The bits in the stridemask value represent the Least Significant Bit of the stride. Thus a value of 0x4000 calls for a stride of 16384 (using Bit 14 of the address). Mulitple bits can be set mixing row and column strides. The Memory Controller section of the PRM for the CPU of the test system for information on how the memory reference address is divided between rows and columns in the DRAM. The value can be specified as a decimal (NNN), hexadecimal (0xNNN), or octal (0NNN) value. The maximum value is 0x400000 (4194304). The default value is 0xC600 which represents strides using Bits 15, 14, 10, and 9. Default values specific to UltraSPARC T1 and UltraSPARC IIIi based systems are as follows: • `ram.h:#define DSTRIDEMASK 0xC600` • `ram.h:#define DSTRIDEMASKFIESTA 0x1C040` • `ram.h:#define DSTRIDEMASKNIAGARA1 0x180C00`
Memory Locking	By default memory locking is Disabled. To turn it on, set lock to Enabled. This test uses ISM to lock the memory into the core, which allows 4 MB virtual pages and avoids swapping. Running without locking, adds more randomness to the addressing sequence. If memory locking with ISM fails, the test allocates on heap and tries to lock the memory allocated on heap.
ECC Error Monitor	ECC Monitor is Enabled by default. The ECC error monitor runs as a separate thread in the test. When an ECC error is detected, the message is displayed in the test output. Turn off the monitor by setting this option to Disabled.
ECC Error Threshold	The number of ECC errors after which the test will stop (if ECC monitor is running). When the threshold is reached, the test exits with a nonzero exit code. If set to zero, the test will still report all the errors but will not stop. The default of threshold is 2.
Number of Passes	Specifies the number of passes, in the same instance. Increase passes in case lock is enabled. This saves time spent on locking the memory every time a new process or instance is spawned by the SunVTS kernel. This pass has no relation with the system passes in the SunVTS infrastructure. It appears that `ramtest` is taking longer to complete system passes.

TABLE 48-1 `ramtest` Options *(Continued)*

`ramtest` Options	Description
NTA March Test	Specifies number of loops of NTA March(30N) Test, per pass. Increasing the number of loops of any subtest increases the relative time spent on that subtest in each pass. This increase also increases the time taken to complete a pass. NTA March Test attacks coupling and stuck at faults. NTA March is efficient at finding single-, double-, and some triple-bit errors. Depending on the stride option, coupling faults can be found between cells in adjacent columns, or rows that are targeted. Test time will be higher when row striding is selected because of greater page faults generated. For efficiency purposes, total memory is divided among available CPUs.
LA March Test	Specifies number of loops of LA March(22N) Test, per pass. Increasing the number of loops of any subtest increases the relative time spent on that subtest in each pass. This increase also increases the time taken to complete a pass. LA March Test attacks coupling and stuck-at-faults.
LR March Test	Specifies number of loops of LR march(14N) test, per pass. Increasing the number of loops of any subtest increases the relative time spent on that subtest in each pass. This increase also increases the time taken to complete a pass. LR March Test attacks coupling and stuck-at-faults.
SS March Test	Specifies number of loops of SS March(22N) Test, per pass. Increasing the number of loops of any subtest increases the relative time spent on that subtest in each pass. This increase also increases the time taken to complete a pass. SS March Test attacks simple static faults.

ramtest Test Modes

TABLE 48-2 ramtest Supported Test Modes

Test Mode	Description
Exclusive	Stresses memory modules and generates enormous amount of memory traffic.

ramtest Command-Line Syntax

/opt/SUNWvts/bin/sparcv9/ramtest *standard-arguments* **[-o**

[bgpattern=*Disabled/Solid/Checkerboard/RowStripe/ColumnStripe/Random/Randexcl*
[reserve=*Integer between 0 and 90* **]**
[stride=*Row* | *Column* | *Random* | *UserDefined* | *Custom*
[userstride=*1 - 16* **]**
[stridemask=*0x40 - 0x400000*]
[lock=*Enabled* | *Disabled* **]**
[dratio=*Integer between 0 and 100* **]**
[eccmonitor=*Enabled* | *Disabled* **]**
[threshold=*Integer i; 0 = i = MAX-INT* **]**
[pass=*32 bit integer* **]**
[ntaloops=*32 bit integer***]**
[laloops=*32 bit Integer* **]**
[lrloops=*32 bit Integer* **]**
[ssloops=*32 bit Integer*]]

TABLE 48-3 `ramtest` Command-Line Syntax

Argument	Description
bgpattern	Enhanced for UltraSPARC T1 (high-end processor with chip multithreading [CMT]) based systems to use background patterns to work as the base patterns for read/write operations within the march test selected. This technique has proven to be very effective and useful for l2sram tests that use similar pattern generation techniques. You can, however, still use the previous methods.
reserve	Specifies the amount of memory that will not be allocated for testing. `reserve` represents a percentage of the total physical memory in the system. When the test starts, it probes the total memory present in the system, then tries to allocate (100 - `reserve`)% of memory. If the allocation or locking does not succeed the amount of memory is reduced before the retry. Before starting the test, the amount of memory allocated for testing is displayed. Default value for `reserve` option is 20. For UltraSPACR IIIi platforms, the default value is 25. On low memory systems, keep the reserve value higher to avoid excessive swapping. For 32-bit booted systems, the reserve value represents the percentage of 4 GB rather than the percentage of total physical memory.
stride	By default `stride` is set to `random`. It can be set to Column or Row also. For `random`, either Row or Column are randomly selected for each pass. The value of `stride` defines the memory locations addressed consecutively in certain subtests, in a hardware dependent manner. All testable memory is still tested. Using different `stride` checks coupling among a different set of memory cells, therefore random is the recommended value for this option unless both Column and Row are being explicitly used in different instances. For FA type of uses, stride may also be set to UserDefined, in this case the test will stride the number of banks specified in the `userstride` option. `stride` may be set to Custom in which case the `stride` values are randomly selected from the strides specified in the `stridemask` value.
userstride	Use this option to set number of banks the test should stride. One of the good choices could be the interleave on the suspect bank, during FA. the value is limited between 1 and 16. This also means row striding is not possible while using this option.

TABLE 48-3 `ramtest` Command-Line Syntax

Argument	Description
`stridemask`	When `stride=custom` is selected, this value specifies the strides used. Each thread selects one of the stride values from `stridemask` by selecting one of the bits in the mask. The bits in the `stridemask` value represent the Least Significant bit of the stride. Thus a value of 0x4000 calls for a stride of 16384 (using bit 14 of the address). Mulitple bits can be set by mixing row and column strides. The value can be specified as a decimal (NNN), hexadecimal (0xNNN), or octal (0NNN) value. The maximum value is 0x400000 (4194304). The default value is 0xC600 which represents strides using Bits 15, 14, 10, and 9.
`lock`	By default memory locking is disabled. To turn it on set the lock to enabled. The test uses ISM to lock the memory into the core, this gives 4 MB virtual pages and avoids swapping. Running without locking adds more randomness to the addressing sequence. If memory locking with ISM fails, the test allocates on heap and tries to lock the memory allocated on heap. On low memory systems, this option can be enabled to avoid excessive swapping. Solaris 10 users, perform the following steps: 1. Issue the following command: % **`prctl $$`** If resource controls `project.max-shm-memory` and `project.max-shm-ids` are listed in the output, proceed to the next step, otherwise follow the instructions given for Solaris 9. 2. Retrieve the default project with the following command: % **`projects -d root`** This command outputs the default project name, *project1* in this example, for the Super User. 3. Set the resource control `project.max-shm-memory` with the following command: % **`prctl -t privileged -r -n \`** **`project.max-shm-memory -v 9223372036854775807 \`** **`-i project`** *project1* For further information please refer to the *Solaris Tunable Parameters Reference Manual* applicable to your Solaris release.
`eccmonitor`	ECC Monitor is enabled by default. The ECC error monitor runs as a separate thread in the test. When an ECC error is detected, the message is displayed on to the test output. The monitor can be turned off by setting this option to disabled.

TABLE 48-3 `ramtest` Command-Line Syntax

Argument	Description
`threshold`	Number of ECC errors after which the test stops (if ECC monitor is running). When the threshold is reached the test will exit with a non zero exit code. If set to zero, the test will still report all the errors but will not stop. The default threshold is 2.
`pass`	Specifies the number of passes in the single instance. Increase `pass` if `lock` is enabled. This saves time spent on locking the memory when a new process or instance is spawned by the SunVTS kernel. This `pass` has no relation with the system passes in the SunVTS infrastructure, it will appear that `ramtest` is taking longer to complete system passes.
`ntaloops`	Specifies number of loops of NTA march(30N) test, per pass. Increasing the number of loops of any subtest increases the relative time spent on that subtest in each pass. This increase also increases the time taken to complete a pass. NTA march test attacks stuck-at-faults, two-cell coupling faults, and some three-cell coupling faults.
`laloops`	Specifies number of loops of LA march(22N) test, per pass. Increasing the number of loops of any subtest increases the relative time spent on that subtest in each pass. This increase also increases the time taken to complete a pass. LA march test attacks coupling and stuck-at-faults.
`ntaloops`	Specifies number of loops of NTA march test, per pass. Increasing the number of loops of any subtest increases the relative time spent on that subtest in each pass. This increase also increases the time taken to complete a pass. NTA march test attacks coupling and stuck at faults.
`lrloops`	Specifies number of loops of LR march(14N) test, per pass. Increasing the number of loops of any subtest increases the relative time spent on that subtest in each pass. This increase also increases the time taken to complete a pass. LR march test attacks coupling and stuck-at-faults.

TABLE 48-3 `ramtest` Command-Line Syntax

Argument	Description
`dratio`	(Descramble ratio) Tunes the algorithm used to generate data patterns in `ramtest`. A descramble ratio of 100 means that all the data patterns generated will be descrambled. If a descramble ratio is 0, the test will generate the data patterns tuned toward bus noise. Default value is 50, which means that half the data patterns are descrambled.
`ssloops`	Specifies number of loops of SS march(22N) test, per pass. Increasing the number of loops of any subtest increases the relative time spent on that subtest in each pass. This increase also increases the time taken to complete a pass. The SS March test attacks simple static faults.
`custom`	When `stride=custom` is selected, this value specifies the strides used. Each thread selects one of the stride values from `stridemask` by selecting one of the bits in the mask. The bits in the `stridemask` value represent the least significant bit of the `stride`. Thus a value of 0x4000 calls for a `stride` of 16384 (using Bit 14 of the address). Mulitple bits can be set mixing row and column strides. The value can be specified as a decimal (NNN), hexadecimal (0xNNN), or octal (0NNN) value. The maximum value is 0x400000 (4194304). The default value is 0xC600 which represents strides using Bits 15, 14, 10, and 9.

Note – 32-bit tests are located in the bin subdirectory, `/opt/SUNWvts/bin/`*testname*.

Note – On the Solaris 10 OS, ECC errors are logged in fault management architecture (FMA) error logs by the FMA subsystem of the OS.

CHAPTER 49

Serial Asynchronous Interface (PCI) Test (saiptest)

saiptest Description

The saiptest checks the functionality of the serial asynchronous interface (SAI) card through its device driver.

Note – You must have Patch ID 109338 installed on the system where you plan to run the saiptest.

---

saiptest Requirements

Before running the SunVTS diagnostics software, ensure that you install the device driver and the cards to be tested. Also, reboot your system with the boot -r command to reconfigure the system and enable the SunVTS kernel to recognize the new driver.

Note – You must run the `saiptest` in Intervention mode.

Note – You must have Patch ID 109338 installed on the system where you plan to run the `sapitest`.

The following minimum hardware configuration is required to successfully run the internal test:

- PCI-based SPARC system with a PCI slot
- Serial asynchronous interface card, installed in one of the PCI slots

The following hardware is also required to run other SunVTS Serial Asynchronous Interface tests:

- Serial asynchronous interface patch panel (part no. 370-2810)
- 25-pin serial loopback plugs (part no. 540-1558)
- RS-232 serial cables (part no. 530-1685)
- TTY terminal

`saiptest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

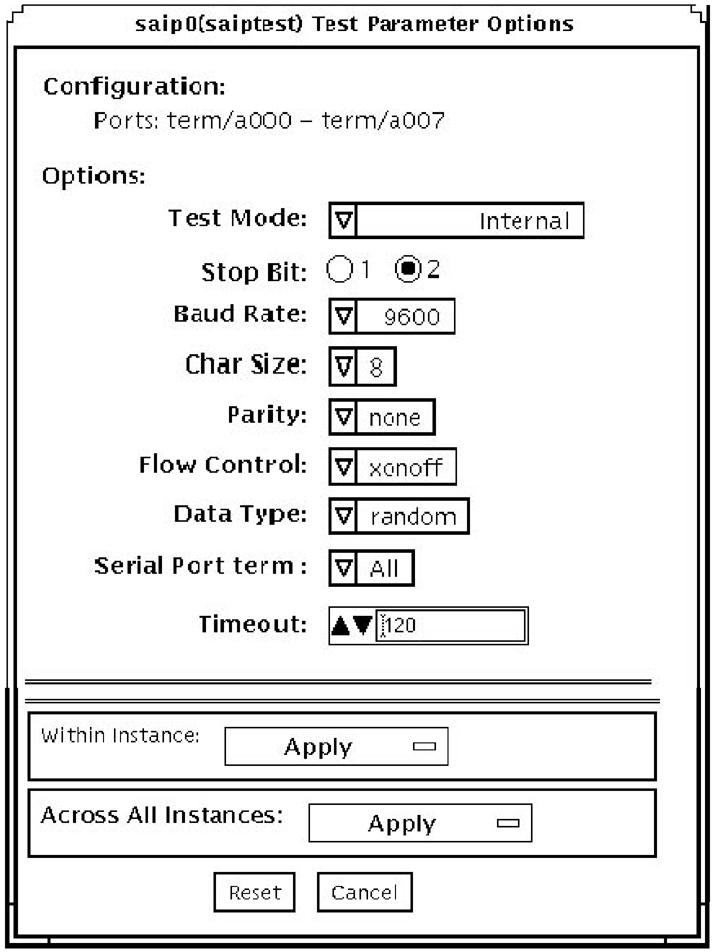


FIGURE 49-1 `saiptest` Test Parameter Options Dialog Box

The Configuration section of the Options dialog box displays the asynchronous serial ports available for the SAI board. The following table shows the available ports.

TABLE 49-1 `saiptest` Asynchronous Serial Ports

Board Number	Board Device	Serial Ports
0	saip0	term/a000-a007
1	saip1	term/b000-b007
2	saip2	term/c000-c007

TABLE 49-1 `saiptest` Asynchronous Serial Ports *(Continued)*

Board Number	Board Device	Serial Ports
3	saip3	term/d000-d007
4	saip4	term/e000-e007
5	saip5	term/f000-f007
6	saip6	term/g000-g007
7	saip7	term/h000-h007
8	saip8	term/i000-i007
9	saip9	term/j000-j007
10	saip10	term/k000-k007
11	saip11	term/l000-l007
12	saip12	term/m000-m007
13	saip 13	term/n000-n007
14	saip 14	term/o000-o007
15	saip 15	term/p00-p007

TABLE 49-2 `saiptest` Options

`saiptest` **Option**	**Description**
Test Mode: Internal	Performs internal loopback testing on the SAI card(s) installed in PCI slots. You do not need to attach anything to the card(s) to perform this test.
Test Mode: 25-pin loopback	Provides full-duplex transmission and full-modem loopback testing of the serial port selected in the Serial Port section of the dialog box. Attach the 25-pin loopback plug to the serial port on the serial asynchronous interface patch panel that is being tested. This test cannot be run concurrently with the Echo-TTY option enabled.
Test Mode: Echo-TTY	Checks the proper operation of the serial port selected in the Serial Port section of the option by echoing characters typed on a TTY terminal keyboard to the TTY terminal screen. The characters you type on your TTY keyboard display on the TTY screen. Note: A TTY connection to the serial asynchronous interface serial port requires corresponding character size up. For example, if a TTY attachment is running with 8-bit character size, then the Char Size `saiptest` option should be set to 8- bits. If you do not type any characters within two minutes, this test times out.

TABLE 49-2 `saiptest` Options *(Continued)*

`saiptest` **Option**	**Description**
Baud Rate	Specifies the baud rate. Choose 110, 300, 600, 1200, 2400, 4800, 9600, 19200, or 38400 baud. Note: The baud rate of 38400 can only be used if the Internal test is disabled and you are testing one port at a time.
Char Size	Specifies the character length. Choose 5, 6, 7, or 8 characters.
Stop Bit	Specifies the number of stop bits. Choose 1 or 2 bits.
Parity	Specifies the selectable parity. Choose none, odd, or even.
Flow Control	Specifies the selectable flow control. Choose XOnOff, rtscts, or both.
Data Type	Specifies the selectable data type pattern. Choose 0x55555555 (0x55), 0xaaaaaaaa (0xaa), or random.
Serial Port	Specifies the serial port to be tested. The available ports are listed in the Configurations section at the top of the `saiptest` dialog box.
Timeout	Specifies the number of seconds until the test times out. The default is 120 seconds.

`saiptest` Test Modes

TABLE 49-3 `saiptest` Supported Test Modes

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

`saiptest` Command-Line Syntax

/opt/SUNWvts/bin/saiptest *standard-arguments* **-o dev=***device-name***, M=***test-mode***,B=baud-rate,Size=***character-size***,Stop=***n-of-stop-bits***, Parity=***parity***,F=***flow-control***,Data=***test-pattern***,sp=***serial-port***,tout=***time-out*

TABLE 49-4 `saiptest` Command-Line Syntax

Argument	Description
dev=*device-name*	Specifies the asynchronous serial ports of the PCI card slots tested. Since there is no default, you must type a device name—either a board (saip0-16) or an individual port (term/x000-term/x007, where x is a-p): •saip0 = the 8 asynchronous serial ports in the first card •saip1 = the 8 asynchronous serial ports in the second card •saip2 = the 8 asynchronous serial ports in the third card •saip3 = the 8 asynchronous serial ports in the fourth card •saip4 = the 8 asynchronous serial ports in the fifth card •saip5 = the 8 asynchronous serial ports in the sixth card •saip6 = the 8 asynchronous serial ports in the seventh card •saip7 = the 8 asynchronous serial ports in the eighth card •saip8 = the 8 asynchronous serial ports in the ninth card •saip9 = the 8 asynchronous serial ports in the tenth card •saip10 = the 8 asynchronous serial ports in the eleventh card •saip11 = the 8 asynchronous serial ports in the twelfth card •saip12 = the 8 asynchronous serial ports in the thirteenth card •saip13 = the 8 asynchronous serial ports in the fourteenth card •saip14 = the 8 asynchronous serial ports in the fifteenth card •saip15 = the 8 asynchronous serial ports in the sixteenth card or `/dev/term/x00`*m* Where *x* is a-p and m is 0 to 7 (any of the asynchronous serial ports in PCI card slots).
M=*test-mode*	Specifies Internal, 25-pin-loopback, or Echo-TTY test mode.
B=*baud-rate*	Sets the baud rate to 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400. The default is 9600.
Stop=*#of-stop-bits*	Toggles the number of stop bits between 1 or 2. The default is 1.
Size=*character-size*	Sets the character size as an integer between 5 and 8.
Parity=*parity*	Specifies the parity as `none`, `odd`, or `even`. The default is none.
F=*flow-control*	Specifies flow control as `xonoff`, `rtscts`, or both.
Data=*test-pattern*	Specifies test pattern as 0x55555555, 0xAAAAAAAA, or random.
sp=*serial-port*	Specifies the terminal and asynchronous serial port number, such as term/a00*n* (sp=*n*).
tout=*time-out*	Specifies the number of seconds until the test times out. The default is 120 seconds.

Note – The `saiptest` error messages are generated when the SunVTS Serial Asynchronous Interface discovers errors. The error descriptions that appear in the VTS interface identify probable causes for the card or test failure, and identify the field replaceable unit (FRU) and recommended action, if possible. The two FRUs under test are: the Serial Asynchronous Interface card, and the patch panel with the cable attached.

CHAPTER **50**

Internal I²C Smart Card Reader Test (sc2test)

sc2test Description

The sc2test verifies the proper functioning and integrity of the internal I²C smart card reader by testing the scmi2c(7d) driver.

sc2test Subtests

The sc2test consists of the following subtests:

- Register subtests:
  - Register read subtest

    The Register read subtest performs read access to selected scmi2c reader registers by means of ioctl().
  - Walking 1s subtest

    The Walking 1s subtest performs walking 1s to selected scmi2c reader registers; registers under test are saved and restored.
- AnswerToReset (ATR) subtest

The sc2test initializes the card reader. The test power manages and resets the card. After reset, the test makes multiple attempts to read the ATRs from the inserted card to verify against a list of recognized ATRs. The cards supported are the Cyberflex and Payflex cards are supported.

- APDU Unique-ID subtest

  Based on the result of the ATR, the Application Protocol Data Unit (APDU) Unique-ID subtest exchanges selected ISO 7816 APDUs between the card and the reader to retreive the card's unique ID.

Caution – The sc2test cannot run when the ocfserv smart card server application is running, because the ocfserv has exclusive open on /dev/scmi2c0. The sc2test detects the ocfserv process running and prompts you to kill ocfserv. Also, the sc2test cannot run if any other third-party smart card host application has exclusive open on /dev/scmi2c0 device.

Caution – The file /opt/SUNWvts/bin/sc2test.atr consists of an ATR list for the supported Cyberflex or Payflex cards. You can use an editing program to update the file to include the new ATR.

---

sc2test Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

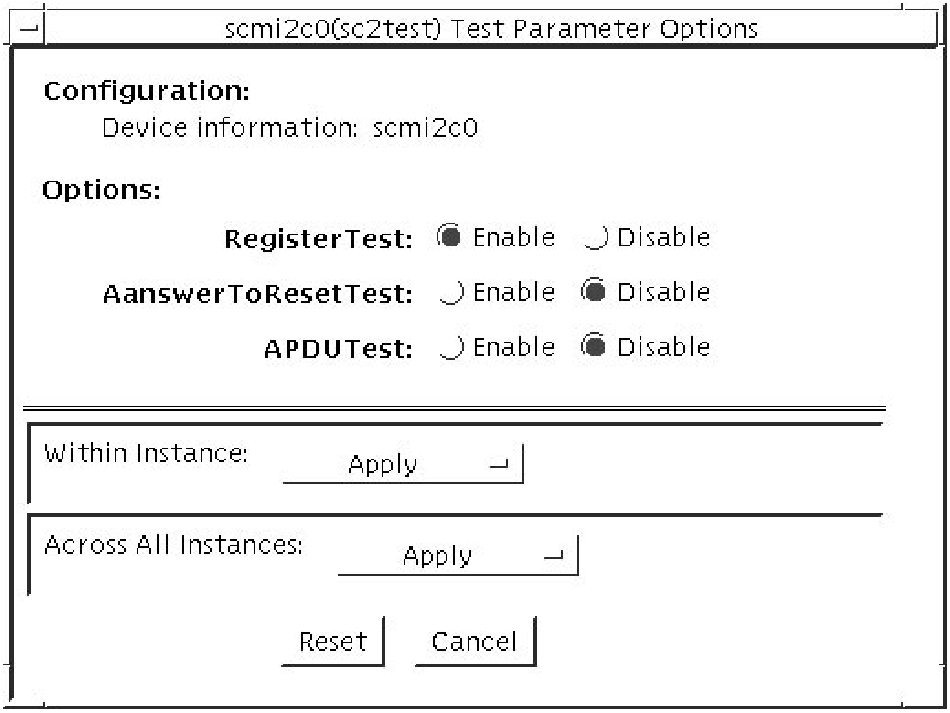


FIGURE 50-1 `sc2test` Test Parameter Options Dialog Box

TABLE 50-1 `sc2test` Options

`sc2test` **Options**	**Description**
Register Test	Enables or disables the Register subtests. The default is Enable.
AnswerToReset Test	Enables or disables the AnswerToReset subtest. The default is Disable.
APDU Test	Enables or disables the APDU subtest. The default is Disable.

sc2test Test Modes

TABLE 50-2 sc2test Supported Test Modes

Test Mode	Description
Connection	Performs the Register read and Walking 1s subtests.
Functional	Performs the Register subtests, the AnswerToReset subtest, and the APDU unique-ID subtest.

sc2test Command-Line Syntax

/opt/SUNWvts/bin/sc2test *standard-arguments* **-o [**
[dev=*device-name* **]**
[regs=*Enable|disable* **]**
[atr=*Enable|disable* **]**
[apdu=*Enable|disable* **]]**

TABLE 50-3 sc2test Command-Line Syntax

Argument	Description
dev=*device-name*	`device-name` is the logical device name to be tested, for example, `dev=scmi2c0`.
reg=*Enable|disable*	Enables or disables the Register tests. The default is Enable.
atr=*Enable|disable*	Enables or disables the ATR test. The default is Disable.
apdu=*Enable|disable*	Enables or disables the APDU unique-id test. The default is Disable.

CHAPTER 51

SEEPROM Test (seepromtest)

seepromtest Description

seepromtest verifies the functionality of the SEEPROM. The probing portion of seepromtest traces the PICL (Platform Information and Control Library) tree and finds the SEEPROM nodes with physical addresses. In addition, the probing portion of seepromtest verifies the size of the physical parent of the SEEPROM. Once the probing portion is complete, seepromtest reads each byte of the SEEPROM devices to verify that the SEEPROM is the correct size. Finally, the seepromtest checks the read operation of the SEEPROM devices.

Note – seepromtest can only be performed on platforms with one or more SEEPROM physical address in the PICL tree. Currently, seepromtest is only supported on the Sun Blade 1000/2000 workstations.

---

seepromtest Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

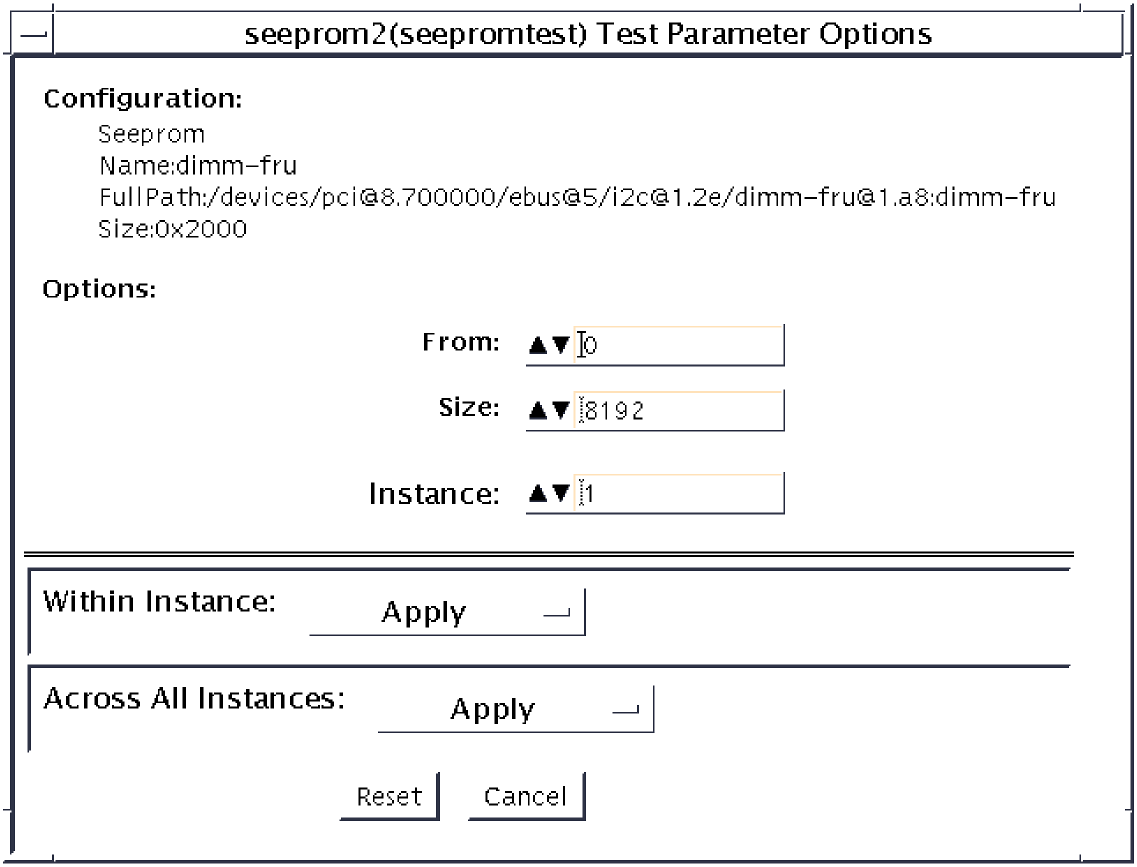


FIGURE 51-1 `seepromtest` Test Parameter Options Dialog Box

TABLE 51-1 `seepromtest` Options

`seepromtest` **Options**	**Description**
From	Specifies the number from which the offset of the SEEPROM is read.
Size	Specifies the number of bytes to be read from the SEEPROM.

`seepromtest` Test Modes

TABLE 51-2 `seepromtest` Supported Test Modes

Test Mode	Description
Connection	Performs the entire `seepromtest`.
Functional	Performs the entire `seepromtest`.

`seepromtest` Command-Line Syntax

`/opt/SUNWvts/bin/seepromtest` *standard_arguments* **`[-o`**
`[ dev=`*device_name* **`]`**
`[ from=`*number* **`]`**
`[ size=`*number* **`] ]`**

TABLE 51-3 `seepromtest` Command-Line Syntax

Argument	Description
`dev=`*device-name*	The name of the SEEPROM device, for example, `seeprom0`.
`from=`*number*	The number from which the offset is read.
`size=`*number*	The number of bytes to be read.

SEEPROM devices do not have device names. `seepromtest` assigns a device name such as `seeprom0`, `seeprom1`, `seeprom2`. If you do not specify a device name in the command-line syntax, the `seepromtest` lists the device names as `seeprom0`, `seeprom1` or `seeprom2`, and the physical addresses.

When invoked, `seepromtest` displays a list of the SEEPROMs in the system and their usage. The following is an example of invoking `seepromtest` without any arguments. Notice the logical names on the left and physical names on the right.

```
seeprom0 : /devices/.../dimm-fru@1,a0:dimm-fru
seeprom1 : /devices/.../dimm-fru@1,a4:dimm-fru
...
```

CHAPTER 52

Serial Ports Test (serialtest)

serialtest Description

The `serialtest` checks the system on-board serial ports (`su[0,1]`, `zs[0,1]`, `zsh[0,1]`, `se[0,1]`, `se_hdlc[0,1]`, `asy[0,1]`), and any multiterminal interface (ALM2) boards (`mcp[0-3]`). Data is written and read in asynchronous and synchronous modes using various loopback paths.

Note – `serialtest` supports testing the `asy` driver (16550 UART). The 16550 device only works in asynchronous mode. The maximum baud rate is 115200.

Note – The 16550 UART can support two devices but some systems only carry one physical port. Do not try to test port a. The operating system shows two ports, but physically there is only one port.

Intervention mode must be enabled to run this test. This test is nonscalable.

Loopback Connectors

The loopback test requires null modem and plug connectors, which are described in Appendix A.

There are a variety of loopback paths available. The internal loopback paths do not require an external connector. Their availability depends on the device. The zs(7D) device has an internal path for synchronous mode and the se(7D) device has an internal path for asynchronous mode. The exact type of loopback connector required depends on the system I/O panel.

The loopback for the Null Modem Ato B option is a female-to-female plug. Its pin configuration is the same as the one described for the section "9-Pin to 9-Pin Port-to-Port Loopback Cable" on page 405.

The loopback for the Plug A to A option is described in the section "9-Pin Female Single-Port Loopback Plug" on page 403.

serialtest Synchronous Testing Software Requirements

If you have zs(7D) serial ports on your machine, the synchronous devices might not exist. Look in the /dev directory for zsh (where h=0 and/or 1). If they do not exist, use the following procedure to create them.

▼ To Create Synchronous Devices

1. **Enter the** `add_drv zsh` **command.**

```
# add_drv zsh
```

2. **Enter the `devfsadm` command. Go to the `/dev` directory to check if the device nodes for `zsh` have been created.**

```
# devfsadm
# cd /dev
# ls zsh
zsh
```

3. **If these steps do not work, enter the `rem_drv zsh` command and then repeat Steps 1 and 2.**

```
# rem_drv zsh
```

`serialtest` Asynchronous and Synchronous Testing

Asynchronous Testing

This mode uses the asynchronous communication protocol as described in the `zs`(7D)and `se`(7D) man pages. The `termio`(7I) interface is used for configuring port characteristics.

You can select the loopback paths to use, the amount of data to transfer, and the baud rate to be used for the transfer.

The test writes and reads data through the loopback path and compares the data to the original data. The test first sends a single character. If no errors or timeouts are detected, the rest of the data is simultaneously written and read, then compared for data integrity.

Synchronous Testing

This mode uses the synchronous hdlc-framing interface as described in the `zsh`(7D) and `se_hdlc`(7D) man pages. Data is written and read in checksum-protected packets.

The user can select the loopback paths to use and the clock rate.

The synchronous test runs in three phases:

1. **The first phase looks for activity on the port. If no activity is detected for at least four seconds, the test proceeds to the next phase. If activity is detected,** `serialtest` **exits with an error.**

2. **The second phase attempts to send and receive one packet. If no packets are detected after five attempts, the test exits with an error. If a packet is returned, the result is compared to the original. If the length and content of the packets do not match exactly, the test exits with an error.**

3. **The third phase attempts to send many packets through the loop. Some packet drops are to be expected especially on a heavily loaded system. The test allows a percentage of the packets to be dropped. The user can set the drop tolerance between 0 percent and 99 percent.**

   The default is 20 percent. If the system is extremely busy, the drop tolerance should be increased. Each packet is compared with its original for length and content. If a mismatch is detected, the test exits with an error.

   This mode of testing is not available for devices connected to the super I/O port. It is, however, available for all other supported I/O ports.

`serialtest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

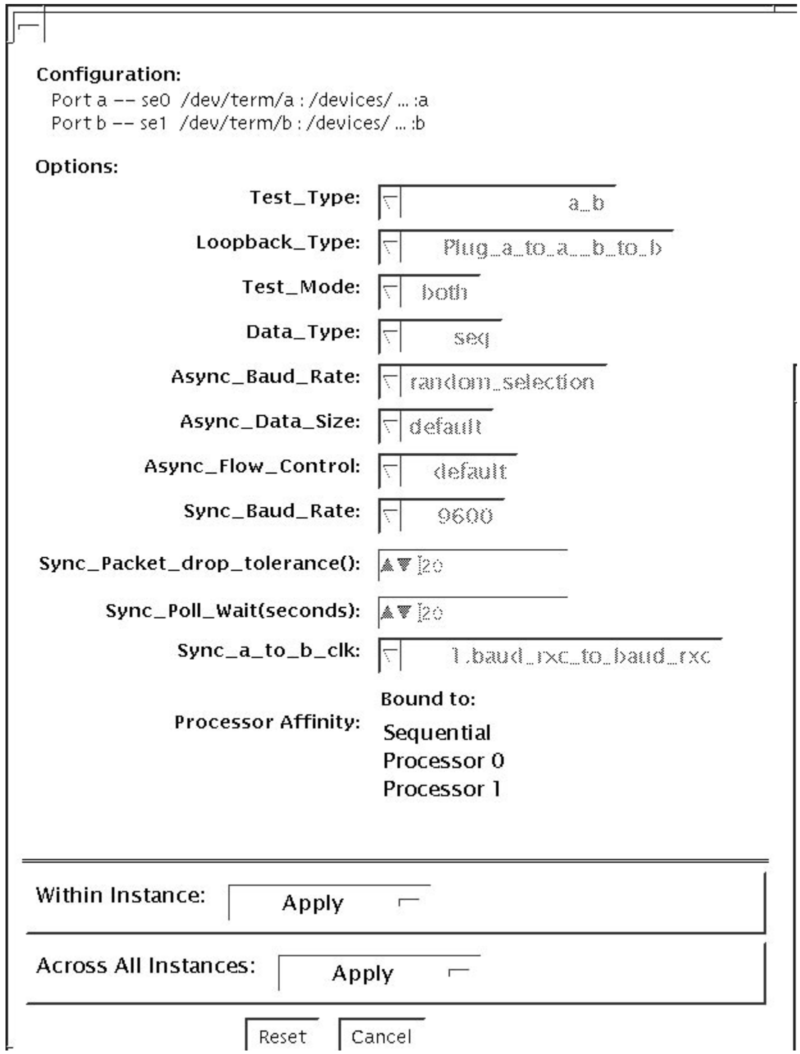


FIGURE 52-1 `serialtest` Test Parameter Options Dialog Box

TABLE 52-1 serialtest Options

Option	Description
Test Type	Selects how the test will run. Test options include: a runs the test on port A. b runs the test on port B. a_b runs the test on ports A and B sequentially. a_b_concurrent runs the test on port A and port B concurrently.
Loopback Type	Selects the loopback test. Options include: Internal is an internal path for a, b, a_b, and a_b_concurrent test types. Plug_a_to_a__b_to_b is an external loopback plug for a, b, a_b, and a_b_concurrent test types. no_modem_a_to_b is an external loopback cable for a_to_b and a_to_b_concurrent test types. Modem_a_to_b is an external loopback cable with a modem attached to generate synchronous Transmit and Receive clocks in synchronous mode. The modem a_to_b external loopback type is intended for Sun internal use only. It requires custom equipment that is not available. (Supported only for SP devices.)
Data Type	Selects the data pattern to transfer. You can select: • Random • Sequential • Alphanumeric • 0x00-0xff
Async Baud Rate	Selects the baud rate for Asynchronous mode testing. The valid rates are 110, 300, 600, 1200, 4800, 9600, 19200, 38400, 57600, 76800, 115200, 153600, 230400, 307200, 460800, and ALL. The default rate is 9600 baud. Some platforms can only support up to 38400 or 76800. If the device is connected to the super I/O port then the max baud rate supported is 115200. The test will return an error if you try to use a higher baud rate than is supported. For baud rates greater than 153600, the serial line drivers must be set for RS-423 mode and not RS-232 mode. The RS-423 and RS-232 modes are usually selected by a hardware jumper on the motherboard. Consult your hardware installation manual for more information.
Async Data Size	Selects the total number of bytes to transfer in Asynchronous mode. This can range from 1 to 10000.
Async Flow Control	Selects the type of flow control to use in Asynchronous mode testing. You can select Hardware (RTS/CTS), Software (XON/XOFF) or None. The default depends on the loopback type. Software flow control is not allowed on a, b, a_b, or a_b_concurrent loopback types.

TABLE 52-1 serialtest Options *(Continued)*

Option	Description
The following options are not available for the devices connected to the super I/O port. They are available for all other supported I/O ports.	
Test Mode	Selects the mode to run the test in. The modes available are Asynchronous, Synchronous, or Both. When Both is selected, the test runs first in Asynchronous mode then Synchronous mode.
Sync Baud Rate	Selects the device generated clock rate for Synchronous mode testing. The valid rates are from 110 to 230400. The rate does not have to be a specific value as required for async mode baud rates. The default rate is 9600. Some platforms can only support up to 38400 or 76800. The test will return an error if you try to use a higher rate than is supported. For rates greater than 100000, the serial line drivers must be set for RS-423 mode and not RS-232 mode. The RS-423 and RS-232 modes are usually selected by a hardware jumper on the motherboard. Consult your hardware installation manual for more information.
Sync Packet	Selects the tolerance level of Synchronous mode dropped packets during the many_packets subtest. The default is Drop20 percent. The valid range is from 0 percent to 99 percent. Some packet drops are expected especially at higher Toleranceclock rates and on a heavily loaded system.
Sync Poll Wait	Selects the number of seconds in additional time to wait for a Synchronous mode packet to be sent. Additional time may be needed when there is heavy system activity and time-outs are being detected. In general, the user can decrease the value to 0 when the system load is light or increase the value when there is a heavy system load.
The following option is available only for devices connected to the super I/O port. They do not apply to any other supported I/O ports.	
User Defined Baud Rate	Allows the user to set new baud rate values for the test. First select the User Defined option from the Async menu. Then enter the new value in the User Defined Baud Rate field.

`serialtest` Test Modes

TABLE 52-2 `serialtest` Supported Test Modes

Mode	Description
Connection	Attempts to open the port to determine if the device is connected. If the port does not open and if the port is not busy, the test exits with an error. If it is successful or fails with a busy or exclusive use error, then the port is considered connected, and the test passes.
Functional	Performs the selected loopback test.

`serialtest` Command-Line Syntax

`/opt/SUNWvts/bin/sparcv9/serialtest` *standard*`-arguments` **`-o dev=`** *device-name*, **`porta=`***port-name*, **`T=`***test-type*, **`L=`***loopback-type*, **`M=`***mode*, **`D=`***data-pattern*, **`AB=`***async-baud-rate*, **`S=`***async-data-size*, **`par=`***none*|*even*|*odd*, **`BS=`** `1|10|100|1000|3000|5000|10000`, **`F=`***flow-control*, **`B=`**`sync-baud-rate`, **`DP=`** *sync-drop-tolerance*, **`P=`***sync-poll-wait*

Command-line syntax for devices connected to the super I/O port:

/opt/SUNWvts/bin/sparcv9/serialtest *standard-arguments* **-o dev=***device-name*, **porta=***port-name*, **T=***test-type*, **L=***loopback-type*, **D=***data-pattern*, **AB=***async-baud-rate*, **BS=***1 | 10 | 100 | 1000 | 3000 | 5000 | 10000*, **F=***flow-control*, **UAB=***User-Baud-Rate*

TABLE 52-3 `serialtest` Command-Line Syntax

Argument	Description
dev=*device-name*	Identifies the serial port(s) to test. There is no default value. You must specify a device name such as: • `se0` • `zs0`, `zs1` • `zs2`, `zs3` • `su0`
porta=*port-name*	The name of the first device of a serial device pair. The default is `A`.
T=*test-type*	Specifies the type of test to run: • a = runs the test on port A. • b = runs the test on port B. • a_b = runs the test on ports A and B sequentially. • a_b_concurrent = runs the test on port A and port B concurrently. • a_to_b = runs the test from port A to port B.
L=*loopback-type*	The type of loopback connector attached to ports: • No_modem_a_to_b • Internal_a_to_a__b_to_b • Plug_a_to_a__b_to_b • Modem_a_to_b (Sun internal only)
M=*mode*	The default test mode is asynchronous. Specify one of the following modes: • asynch • synch • both
D=*data-pattern*	Selects the data pattern to transfer. You can select: • Random • Sequential • Alphanumeric • 0x00-0xFF

TABLE 52-3 `serialtest` Command-Line Syntax *(Continued)*

Argument	Description
AB=*async-baud-rate*	Asynchronous baud rate (default = 9600). The valid values are between 110 - 460800. Note: Some platforms can only support asynchronous baud rates up to 38400 or 76800. For baud rates greater than 153600 the serial line drivers must be set for RS-423 mode and not RS-232 mode.
S=*async-data-size*	Asynchronous mode total number of bytes to write. Values range from 1 to 10000 bytes.
par=*none*|*even*|*odd*	Parity used in the async loop test. Default value is `none`.
BS=*1*|*10*|*100*|*1000*|*3000*|*5000*|*10000*	Number of bytes in each write during async loop test. Default value is 100.
F=*flow-control*	Asynchronous mode flow control: • Hardware (RTS/CTS) • Software (xon/xoff) • None
B=*sync-baud-rate*	Synchronous baud rate (default = 9600). The valid rates are between 110 - 256000. Some platforms can only support synchronous rates up to 38400 or 76800. For rates greater than 100000 the serial line drivers must be set for RS-423 mode and not RS-232 mode.
DP=*sync-drop-tolerance*	Synchronous mode drop packet tolerance (default=20 percent).
P=*sync-poll-wait*	Synchronous mode additional wait time during poll (in seconds).
UAB=*User-Baud-Rate*	Specifies any valid baud rate. Set AB to `u`.

CHAPTER 53

System Service Processor Test (ssptest)

ssptest Description

The ssptest exercises the Remote System Control (RSC) feature, which is integrated on the Sun Enterprise 250 and on the next-generation RSC 2.0 plug-in card introduced with the Sun Fire 280R line. ssptest also exercises the Advanced Lights-Out Management (ALOM) feature, which is integrated into the Sun Fire V210, V240, V215, V245, V440, and V445, and Sun Netra 240 and 440 systems.

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 I^2C bus, and reset lines.

The RSC 1.0 hardware consists of the controller, flash, SEEPROM, 10-MB 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, FRUSEEPROM, Time of Day (ToD) device, Serial Transceiver, and battery backup.

ssptest is not scalable.

ssptest Subtests

The ssptest presents 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 53-1 Subtests for Both RSC 1.0, RSC 2.0, and ALOM

Subtest	Description
Ethernet	Enables internal loopback testing, on the Ethernet device with user specified data, size, and number of packets. Enables 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. Enables 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	Enables internal loopback testing with user-specified data and size on the two internal serial ports. Enables 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 53-2 Subtests for RSC 2.0 Only

Subtest	Description
FRU SEEPROM CRC	Performs a checksum test on the SEEPROM device.

TABLE 53-2 Subtests for RSC 2.0 Only

Subtest	Description
I^2C	Tests the I^2c 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 53-3 Subtests for ALOM Only

Subtest	Description
I^2C	Tests the I^2c bus connection between the host and the ALOM.
ToD	Performs multiple reads to the ToD device and verifies that the time is incrementing.
DOC	Performs the DOC (disk on chip) selftest. This subtest is only enabled on platforms supporting `docselftest`.
FRU SEEPROM CRC	Performs a checksum test on the SEEPROM device. Note: The FRU SEEPROM CRC subtest is not enabled in `ssptest` on platforms using ALOM hardware with ALOM firmware. On these platforms, `i2c2test` performs the checksum test on the SEEPROM device in ALOM hardware.

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

`ssptest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

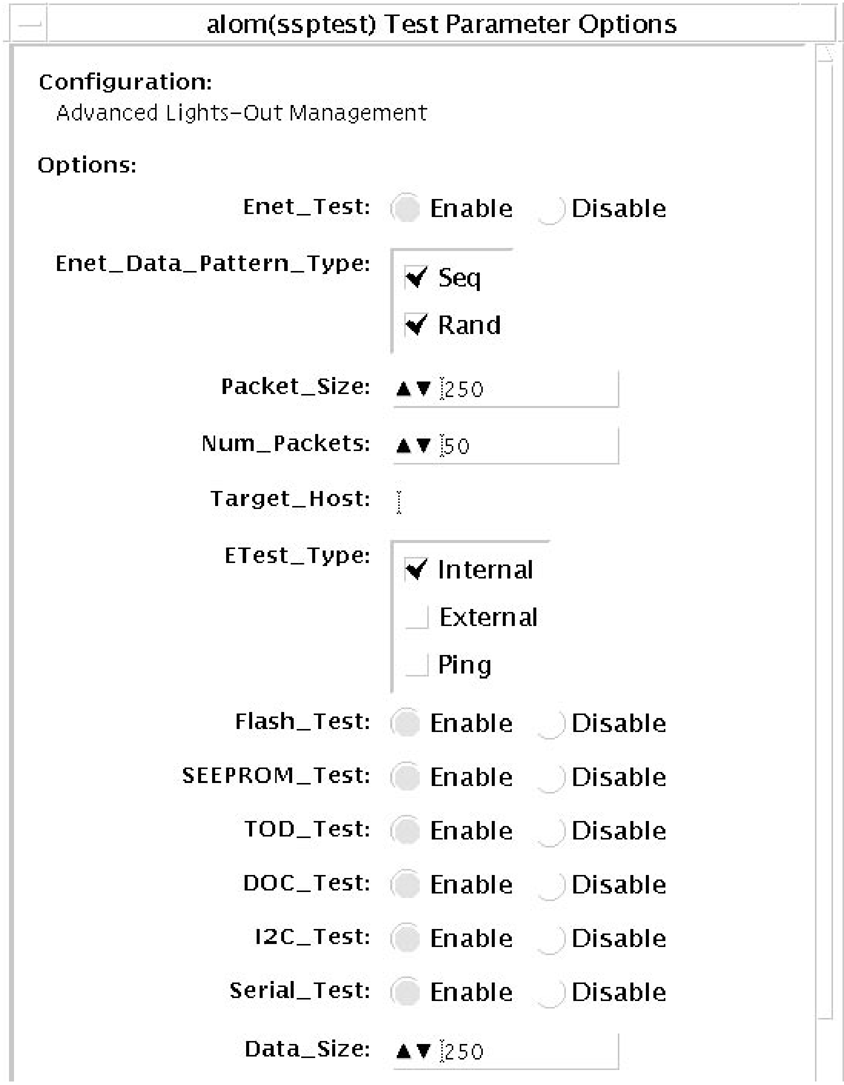


FIGURE 53-1 `ssptest` Test Parameter Options Dialog Box (Top Section)

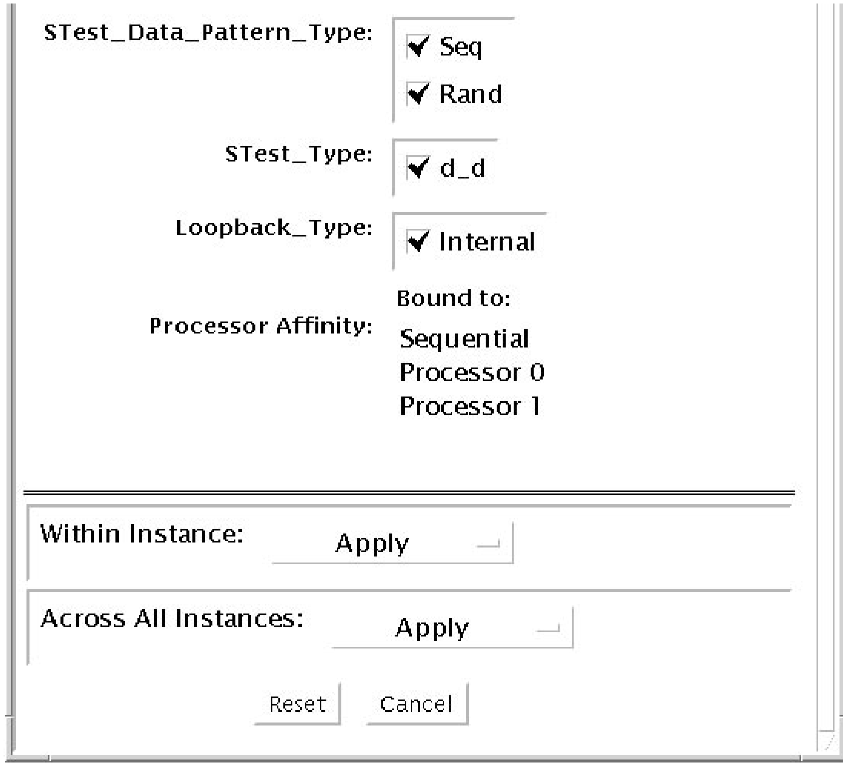


FIGURE 53-2 `ssptest` Test Parameter Options Dialog Box (Bottom Section)

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

TABLE 53-4 `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 53-4 `ssptest` Options *(Continued)*

`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 and ALOM only).
TOD test	Enables or disables the Time Of Day test.
DOC	Enables or disables Disk On Chip test (ALOM only).
I2C test	Enables or disables the I^2C 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	Enables or disables the RSC PCMCIA modem test (RSC 2.0 only).

`ssptest` Test Modes

`ssptest` supports the following test modes as described in the table below.

TABLE 53-5 `ssptest` Supported Test Modes

Test Mode	**Description**
Connection	Reports the status of the RSC or ALOM.
Exclusive	Tests the RSC's and ALOM's Ethernet, flash, SEEPROM, ToD, I^2C, 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***`,doc=`***E/D***`,i2c=`***E/D,***`serial=`** *E/D,***`sdatsize=`***data-size,***`slb=`***I*, **`spattype=`***seq+rand*, **`stest=`***d-d*

TABLE 53-6 `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=`*Enable | Disable*	Enables or disables RSC FRU SEEPROM checksum test (RSC 2.0 and ALOM only).
`tod=`*Enable | Disable*	Enables or disables RSC or ALOM Time of Day test (RSC 2.0 and ALOM only)..
`doc`=*Enable | Disable*	Enables or disables ALOM Disk On Chip test.
`i2c=`*Enable | Disable*	Enables or disables RSC or ALOM I^2c test (RSC 2.0 and ALOM only)..
`serial=`*Enable | Disable*	Enables or disables RSC or ALOM serial test.
`sdatsize=`*data-size*	Data size for the `rsc` or `alom` serial tests.

TABLE 53-6 `ssptest` Command-Line Syntax *(Continued)*

Argument	Description
slb=*Internal+External*	Loopback type. External N/A on ports C and D.
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=*Enable | Disable*	Enables or disables the RSC PCMCIA modem test (RSC2.0 only).

CHAPTER 54

SunHSI Board Test (sunlink)

sunlink Description

The sunlink test verifies the functionality of the SBus and PCI bus SunHSI™ boards by using the HDLC protocol. sunlink initializes and configures the selected channel.

Next, sunlink opens a datagram socket and tries to modify the socket to accept ioctl communications with the driver, and receive synchronous mode information from it.

sunlink then opens the ports, linking the upper and lower layers with ioctl calls. After initialization, this test checks for activity before attempting to send or receive data. An error message is returned if activity is detected. Otherwise the transmit buffer is filled with random data. Random data is used by default. You may also specify other patterns. The data is then transmitted. If the transmission succeeds, sunlink then receives the returned data and verifies that it is identical to what was sent. Finally, statistics about the send and receive are gathered from the socket.

A full sunlink test takes approximately eight minutes per port and makes a brief check of the board ports before the actual test begins. If the port is bad, the test immediately aborts and returns an error message.

`sunlink` Test Requirements

This test will not pass unless you install the correct loopback connectors or port-to-port cables on the ports you are testing. The ports specified for test in the Options dialog box must have loopback connectors attached. See Appendix A for loopback connector part numbers and wiring instructions.

`sunlink` Options

To reach the following dialog box, 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, might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

The Configuration field displays the available ports. (See TABLE 54-1.)

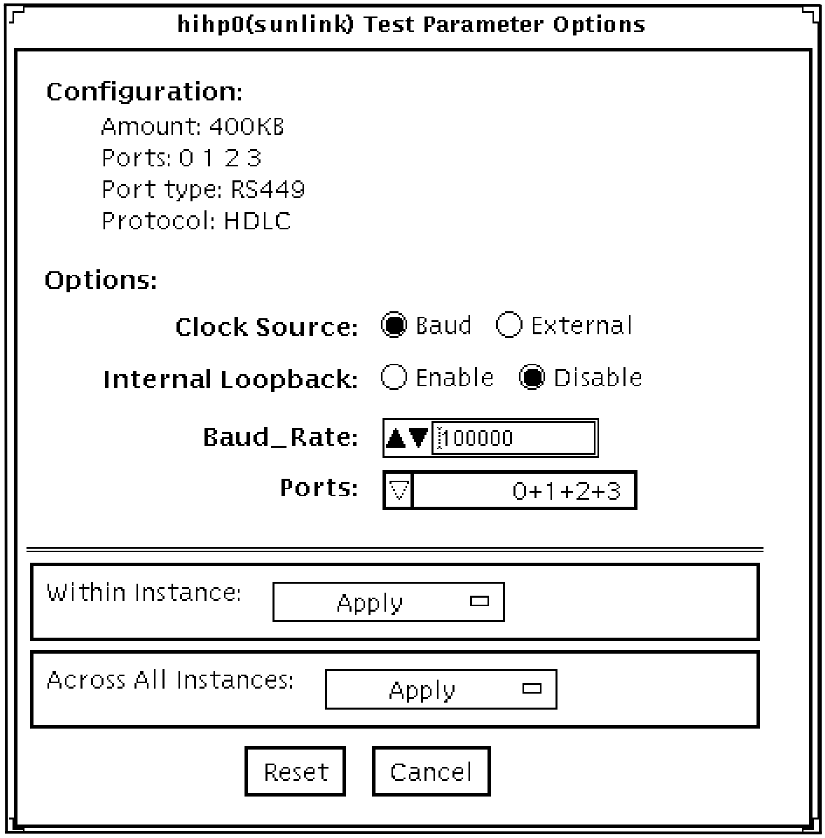


FIGURE 54-1 `sunlink` Test Parameter Options Dialog Box

TABLE 54-1 `sunlink` Options

`sunlink` **Options**	**Description**
Clock source	Selects either the onboard clock or an external clock for use when using `sunlink`. To use the external clock option, the transmit, receive, and clock data lines must be physically looped back.

TABLE 54-1 `sunlink` Options

`sunlink` Options	Description
Internal Loopback	Enables or disables internal loopback tests. Internal Loopback is only needed when the Loopback setting is not port-to-port, and the clock source is onboard.
Baud Rate	Specifies the bit rate transfer speed from 9600 bauds to 2.048 Mb/sec.
Port	Specifies the loopback type—simple single external port loopback, multiple external port loopback, and port-to-port external loopback.

`sunlink` Loopback Connectors

Refer to Appendix A of this manual for information on SunLink™ loopback cables and loopback connectors. Refer to the High Speed Serial Interface hardware manuals for information on null modem cables.

`sunlink` Test Modes

TABLE 54-2 `sunlink` Supported Test Modes

Test Mode	Description
Functional	Runs the full set of tests.

`sunlink` Command-Line Syntax

/opt/SUNWvts/bin/sunlink *standard-arguments* **-o dev=***device-name***,p=***port***,P=***data-pattern***,brate**=*speed-n***,I,C=***clocksource*

TABLE 54-3 `sunlink` Command-Line Syntax

Argument	Explanation
dev=*device-name*	Specifies the device to be tested. Use `hih0` for the HDLC protocol.
p=*ports*	Specifies the port number to be tested.
P=*data-pattern*	Specifies the *data-pattern* as one of the following: • `c`—Character (0x55) • `i`—Incrementing • `d`—Decrementing • `r`—Random (default)
brate=*speed-n*	Specifies the bit rate transfer speed from 9600 bs to 2.048 Mb/sec.
I	Enables internal loopback for HSI.
c=*clocksource*	Specifies the clock source value as one of the following: • `B`—Onboard clock source • `E`—External clock source

The following is typical command-line syntax for testing a SunHSI board:

```
# /opt/SUNWvts/bin/sunlink -o dev=hih0,P=0+1+2+3,brate=100000
```

This command tests the internal loopback for ports 0, 1, 2, and 3. It does not run for the port-to-port internal loopback test.

CHAPTER 55

SunPCi II Test (sunpci2test)

sunpci2test Description

The sunpci2test tests the SunPCi™ II card, which is a PC processor embedded in an add-on card. This test consists of approximately 150 POST routines that perform diagnostic, hardware detection, and initialization functions. This test issues a reset, then launches POST in the SunPCi II BIOS to check the devices. Finally, the sunpci2test runs bridge and system diagnostics tests.

SunPCi-2 and SunPCi-3 cards are tested by the sunpci2test diagnostic. If the card under test is SunPCi-2 then the device name is sunpci2drvX. If the card is SunPCi-3 then the device name is sunpci3drvX.

sunpci2test Test Requirements

Before running the test, the XWindow for Microsoft Windows must be shut down. If this is not done, the test will not launch.

▼ To Shut Down Microsoft Windows and the SunPCi II Card:

1. **Click the Start button in Microsoft Windows.**
2. **Click Shut Down.**

   The shutdown window appears. Wait for the `It is now safe to shut off your PC` message.
3. **Select File from the SunPCi window.**
4. **Select Exit from the file menu.**
5. **Click OK.**

`sunpci2test` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

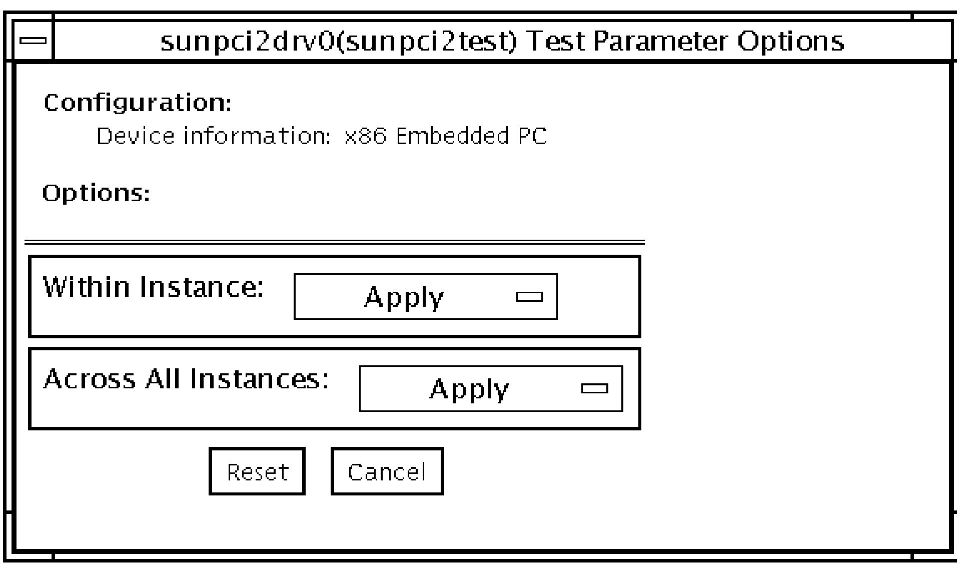


FIGURE 55-1 `sunpci2test` Test Parameter Options Dialog Box

`sunpci2test` only runs with the default parameters in place. The test does not allow any options to be configured specifically for an individual system. The number of instances is preset to 1 (the default value), as only one local copy of the test is supported.

`sunpci2test` Test Modes

TABLE 55-1 `sunpci2test` Supported Test Modes

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

`sunpci2test` Command-Line Syntax

`/opt/SUNWvts/bin/sunpci2test` *standard-arguments*

Note – There are no test-specific options for `sunpci2test`.

CHAPTER **56**

System Test (`systest`)

`systest` Description

The `systest` checks the overall functionality of a Sun system by exercising the CPU, I/O, and memory channels.

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.

Caution – This is an exclusive mode test. This test can not be run in parallel with any other tests or applications.

`systest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

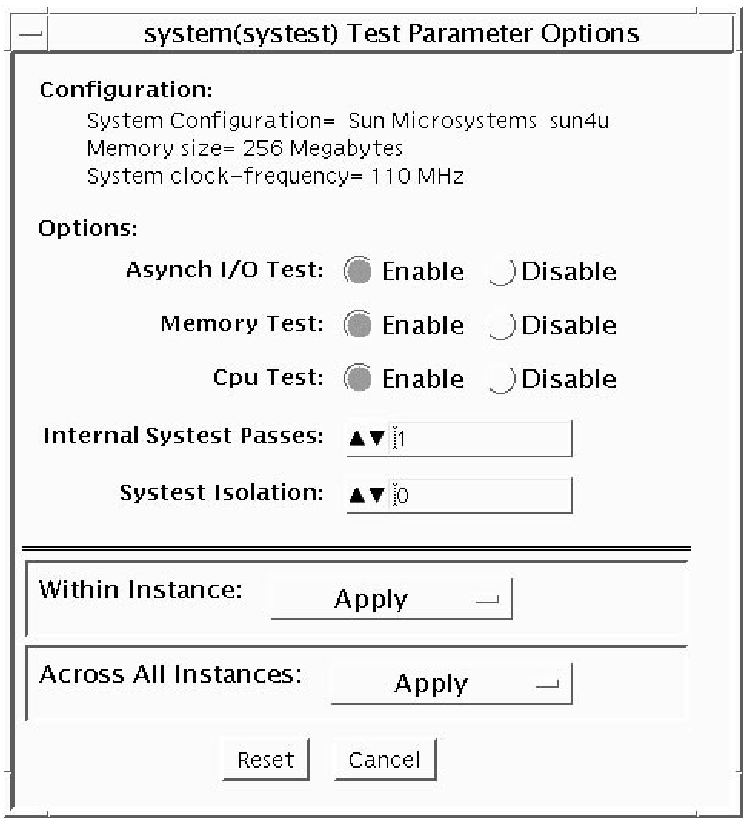


FIGURE 56-1 `systest` Test Parameter Options Dialog Box

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 or 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 56-1 `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

Use the default values for an initial evaluation of the system.

`systest` Test Modes

TABLE 56-2 `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 56-3 `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 is 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

Command-Line Examples

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

Example 1 invokes the following:

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

- `systest` with default parameter values.
- I/O, MEM, and CPU subtests.

Example 2 invokes the following:

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

- `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 or offline CPUs.

Example 3 invokes the following:

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

- 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` performs boards isolation.

Example 4 invokes the following:

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

- 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` performs boards AND CPUs isolation.

CHAPTER 57

Tape Drive Test (`tapetest`)

`tapetest` Description

The `tapetest` synchronous I/O test writes a pattern to a specified number of blocks (or, for a SCSI tape, writes to the end of the tape). The `tapetest` then rewinds the tape and reads and compares the data just written. The `tapetest` asynchronous I/O test sends a series of up to five asynchronous read/write requests to the tape drive, writing to the tape, and then reading and comparing the data. The terms asynchronous and synchronous referred to here, and in the Method field are not related to the SCSI messaging terms of the same name.

The `tapetest` file test writes four files to the tape and then reads them back, comparing the data. For tape library testing, the pass count is incremented only after all tapes in the library have been tested. The read/write algorithms fare enhanced for DLT tape by using a random data pattern (1.5:1 compression) and a more robust read compare algorithm. Some default parameters have also changed.

The Async I/O subtest uses the asynchronous read and write feature of the Solaris tape driver to exercise tape drives. In read-only mode the test sends a maximum of four asynchronous read packets, each with a random size and a random offset, to the tape drive. The test then waits for all outstanding I/O activity to complete before issuing another round of packets. This process continues until the whole area being tested has been covered. In read-write mode, one write packet is issued for every four read packets to ensure a spot check of the write operation. The area of the tape to be tested is written to first in order for the test to work correctly. This test is only supported under the Solaris 8 and Solaris 9 OSs and future compatible releases.

`tapetest` Test Requirements

If you have a tape drive in your system, load a blank writable tape (scratch tape) before you start SunVTS. If you fail to do this, the `tapetest` option may display `drive type:unknown` on the option menu for the `tapetest`.

`tapetest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

`tapetest` supports 4-mm, 8-mm, DLT, 1/4-inch cartridge, and 1/2-inch front-load tape drive testing. The options available for each of the tape devices differ slightly. An example of the Test Parameter Options dialog box for a device is shown in FIGURE 57-1.

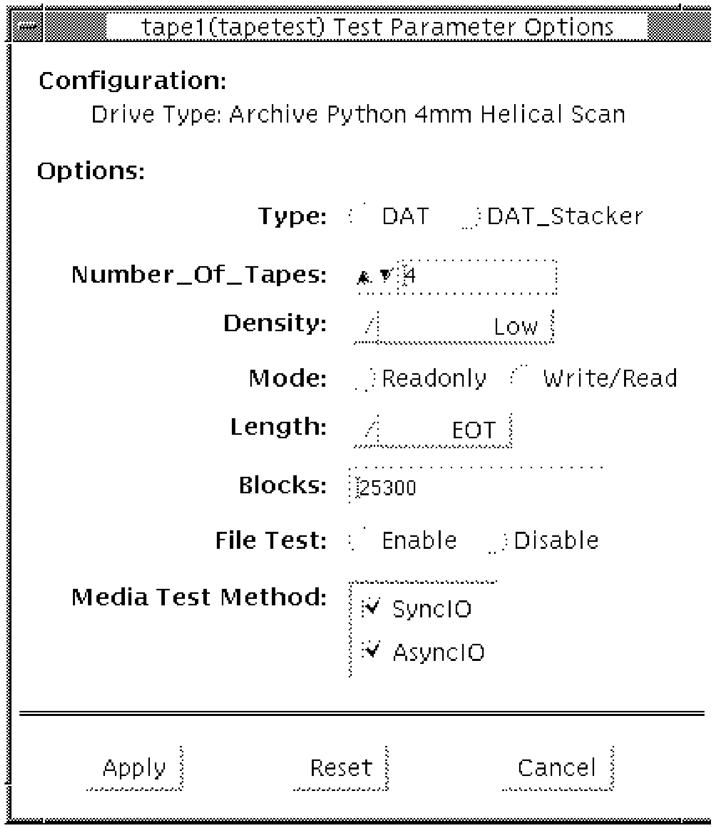


FIGURE 57-1 `tapetest` Test Parameter Options Dialog Box

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

Note – Selecting nondefault options can significantly increase the run time for the test.

Note – The Test Parameter Options dialog box for the 1/4-inch, 1/2-inch, DLT, and 8-mm tape drives differ slightly from FIGURE 57-1.

TABLE 57-1 `tapetest` Options

`tapetest` **Options**	**Description**
Type	Normal tape drive or tape library (stacker).
Number of Tapes	The number of tapes in the tape library. `tapetest` registers a single tape library pass only after all tapes in the library pass.
Density	The following settings are available for most tape drives: • Low – Tests the `l` tape device. • Medium – Tests the `m` tape device. • Compression – Tests the `c` tape device. • All – Tests the `l`, `m`, and `c` tape devices. For half-inch tape drives, the available settings are 800, 1600, and 6250 BPI (blocks per inch). For certain QIC drives, select `QIC-11` (1-byte block ID) mode, `QIC-24` (4-byte block ID) mode, or `Both`. Note: On a DLT drive, the l and m settings both use no compression. `tapetest` does not support changing DLT capacity settings indicated on the front panel.
Mode	If you enable Write/Read mode, the test first writes to the tape and then reads it back to compare. If you enable Read_Only mode, the test assumes the tape has been properly written and merely reads and compares. This mode is useful to check proper head alignment. Note: If a read only pass is attempted and the tape was not previously written by `tapetest`, using the same test parameters currently set, a `Big Read Failure` will occur.
Length	The amount of the tape to be tested. The choices are: • EOT – The default; tests to the entire tape. • Long – The SCSI tape tests 70,000 blocks of the tape. • Short – Only the first 1000 blocks are tested. • Specified – You must type the number of blocks to be tested in the blocks field.
# of Blocks	If you select Specified under the Length option, you must type the number of blocks you want to test.

TABLE 57-1 `tapetest` Options

`tapetest` **Options**	**Description**
Blocksize	Block size specification. This option is only available for Tandberg QIC tape drives. There are two possible values. 512-bytes is for use with older tape media that have transfer size restrictions, while 64-kbytes is for use with current, high-capacity tape media. Note – This option is only available in command line interface mode. Note – With patches 110278-01 or 110211-01 applied, DLT writes either a 512 byte or 65536 byte block depending on how this parameter is set.
File Test	The tape file test sequence is as follows: 1. Writes three files. 2. Rewinds. 3. Reads part of the first file. 4. Forward spaces to the start of the second file. 5. Reads the second file. 6. Forward spaces to the start of the third file. 7. Tries to read to the end of that file for SCSI tapes only. The tape file test tries to backspace to the start of the second file and read it.
Retension	When enable is selected, the program retensions the tape.
Media Test Method	• Sync I/O—`tapetest` reads and or writes the number of blocks selected in Length. • Async I/O—`tapetest` makes four asynchronous read requests to the tape drive. If read and write testing is selected, one asynchronous write request is also sent. The test continues after completing the requests. Note – When testing Tandberg QIC drives, Async I/O testing is restricted to read-only due to asynchronous behavior differences with other tape drives. Note – This option is not associated with the synchronous data transfer request SCSI message. It is only synchronous or asynchronous in nature because the numbers of reads and writes are not synchronous to each other. The SDTR message is not invoked.

`tapetest` Test Modes

The `tapetest` supports the following test modes. It performs different test schemes on the tape device, according to the mode you select.

TABLE 57-2 `tapetest` Supported Test Modes

Test Mode	Description
Connection	`tapetest` verifies that the drive can be opened and that the drive type can be determined. If both checks are successful, or if the drive is currently busy, then the test passes. The `tapetest` fails if the open operation is unsuccessful for any reason other than the drive is busy.
Functional	`tapetest` checks the status, rewinds the tape, erases and retensions it. If the device is a cartridge tape, `tapetest` writes a pattern to `nblks` or `eot` (default), rewinds the tape, and then reads and compares of the pattern. On the other hand, if the device is busy or if no tape cartridge can be found in the drive, the test cannot run and fails.

`tapetest` Command-Line Syntax

`/opt/SUNWvts/bin/tapetest` *standard-arguments* **`-o dev=`***device-name***`,`** **`s=`***block-count***`,d=`***density***`,m=`***mode***`,l=`***length***`,method=`***method***`,ft=`***Enables* | *Disables***`,`** **`ret=`***Enables* | *Disables***`,dat=`***dat-type***`,8mm=`***8mm-type***`,num=`***magazine-size***`,`** **`blocksize=`***block-size*

TABLE 57-3 `tapetest` Command-Line Syntax

Argument	Explanation
`dev=`*device-name*	Specifies the *device-name* of the tape drive (required).
`s=`*block-count*	Specifies the number of blocks to be tested.
`d=`*density*	Specifies the density of the tape to open.
`m=`*mode*	Enables either the Write-Read or Read-Only tests.
`l=`*length*	Specifies the length of the test (EOT, Specified, Long, or Short).
`method=`*method*	Specifies the media test method (SyncI/O and or AsyncI/O) used. Note: This option does not invoke the SCSI message `synchronous data transfer request`. It is only asynchronous or synchronous in nature.

TABLE 57-3 `tapetest` Command-Line Syntax *(Continued)*

Argument	Explanation *(Continued)*
ft=*Enables*|*Disables*	Enables or disables the File test.
ret=*Enables*|*Disables*	Enable or disables tape retension.
dat=*dat-type*	If you are testing a digital audio tape drive, specify whether it is a regular DAT drive or a DAT stacker. The choices are `DAT` and `DAT_Stacker`.
8mm=*8mm-type*	If you are testing an 8-mm tape drive, specify whether it is a regular 8-mm tape drive or a tape library. The command line choices are `8mm` and `8mm_Library`.
num=*magazine-size*	If you are testing a tape library, specify the magazine size.
blocksize= *block-size*	This option is only available on a Tandberg QIC drive and DLT drives. Specify whether to use a 64 kbyte block transfer or a 512 byte block transfer. Use 512 bytes when testing older media in the drive. DLT supports 512 byte and 65536 byte modes

CHAPTER 58

USB Device Test (usbtest)

usbtest Description

The usbtest verifies the proper functioning of the hardware of the USB subsystem. The test supports audio, keyboard, and printer devices. The usbtest for audio verifies the proper functioning of the hardware and software components of the USB audio subsystem. Specifically, the test tests the USB audio devices, USB microphones and speakers.

Note – usbtest tests the USB ports using various USB devices such as the USB key board, the USB printer, or the USB audio devices.

For keyboards, the test verifies whether the keyboard(s) attached to the USB bus are USB-compliant. The test flashes the LEDs of a compliant keyboard. The usbtest for printers verifies the proper functioning and integrity of the USB parallel port devices. The test data, either ASCII or a PostScript file, is sent to the printer through the USB bulk transfer. The test saves and restores the printer settings.

The usbtest options and subtest available depends on the peripheral device attached to the USB port.

usbtest Subtests

XXX lists the `usbtest` subtests for both the USB audio and printer devices.

TABLE 58-1 `usbtest` Subtests

Subtest	Description
USB Audio Devices	
Tone	Supported in Connection test mode only. This subtest generates two seconds of sound output to the speakers. You must decide, by listening, if the test passed or failed.
Record/Play	Records and plays one second of random data at 8kHz sampling. This subtest reads random data from the USB microphone port and plays the music back to the USB speakers.
Audio	Plays a 30 second music file that is output to the speakers. You must decide, by listening, if the test passed or failed. Things to listen for are distortion or lack of music.
USB Printer Devices	
`getdevid`	Retrieves the IEEE 1284 ID string of the printer.
Printer	Prints strings of ASCII characters (from 0x32 to 0x7e), and can also print the postscript file `usbpppdata.ps` to the printer.

usbtest Options

To reach the following 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

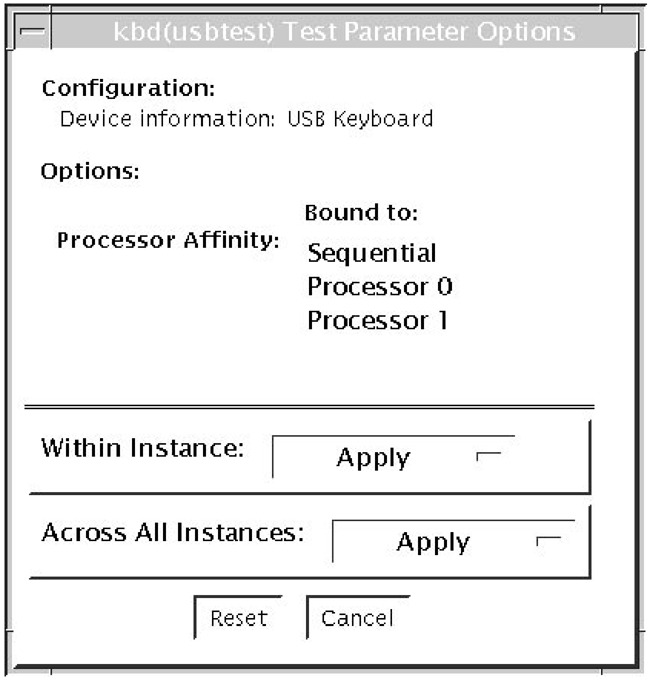


FIGURE 58-1 `usbtest` Test Parameter Options Dialog Box

TABLE 58-2 `usbtest` Opions

Option	Description
USB Audio Devices	
Volume	Specifies the volume of 0 to 255. The default is 80.
Audiotest	Enables or disables the audio music test. The default is Enable.
USB Printer Devices	
DeviceID	Specifies the path to the printer
Printer	Enables or disables the Printer subtest. This test requires a printer to be attached to the USB port.
Printer_data_type	Specifies the test data: ascii, postscript, or both.
Printer_delay_seconds	Specifies, in seconds, the delay after the Printer subtest is performed

`usbtest` Test Modes

This test supports Connection and Functional test modes.

TABLE 58-3 `usbtest` Supported Test Modes

Test Mode	Description
Connection	For a USB audio device, the test outputs two seconds of sound to the speakers For a USB keyboard device, the test runs the full test. For USB printer device, the test runs only the getdevid subtest.
Functional	For a USB audio device, the test performs Records and Plays subtest by default. You can enable Audio Music test. For a USB keyboard device, the test runs the full test. For a USB printer device, the `getdevid` subtest is enabled by default and the Printer subtest is disabled by default.

`usbtest` Command Line Syntax

`/opt/SUNWvts/bin/sparcv9/usbtest -cvf -o dev=`*dev-name*, **`getdevid=`***Enable*|*Disable*, **`printer=`***Enable*|*Disable*, **`data=`***ascii*, **`postscript`**, **`delay=`***0-86400*, **`M=`***Enable*|*Disable*, **`O=`***speaker*|*headphone*|*line-out*, **`V=`***0-255*, **`MF=`***musicfilename*

TABLE 58-4 `usbtest` Command-Line Syntax

Argument	Description
`dev=`*dev-name*	Target device name. Thie test runs on the specified device name. The default device name is `/dev/kbd`.
USB Printer Devices	
`getdevid=`*Enable*|*Disable*	Enables or disables getdevid subtest. This option is available only for a USB printer device. This option is enabled by default.
`printer=`*Enable*|*Disable*	Enables or disables the printer test. This option is disabled by default. This option requires a printer to be attached to the USB port.
`data=`*ascii*	Selects whether to send ASCII text or PostScript data to the printer.

TABLE 58-4 `usbtest` Command-Line Syntax *(Continued)*

Argument	Description
`delay=`*0-86400*	Enables you to choose a delay between passes of the printer test.
USB Audio Devices	
`M=`*Enable* | *Disable*	Enables or disables the music play test. The default is Enable.
`O=`*speaker* | *headphone* | *line-out*	Selects the output device that `usbtest` requires.
`V=`*0-255*	Music output volume. The default is 80.
`MF=`*musicfilename*	Selects the music file. The default is `music.au`.

CHAPTER 59

Virtual Memory Test (vmemtest)

vmemtest Description

The vmemtest checks virtual memory—the combination of physical memory and the swap partitions of the disk(s).

Note – This test might 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 vmemtest 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. Use the `vmemtest` swap space `reserve` option when nonSunVTS test processes are started after SunVTS testing has started. Refer to the *SunVTS User's Guide* for a complete discussion of swap space requirements.

`vmemtest` Options

To reach the following dialog box, 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 might not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

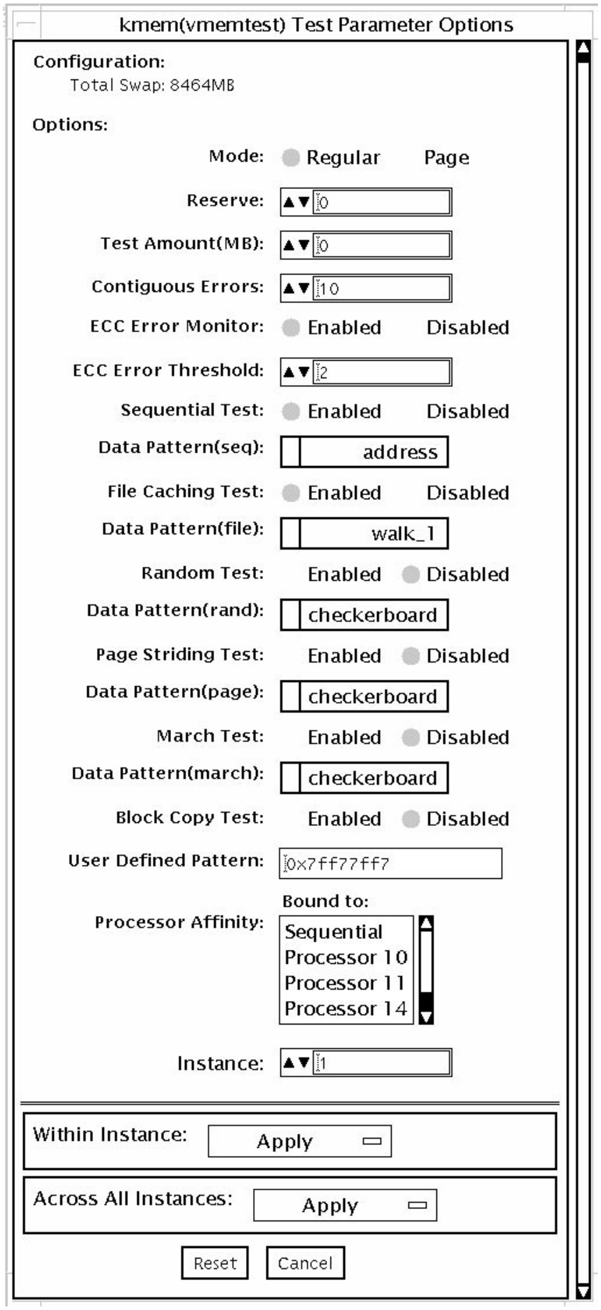


FIGURE 59-1 `vmemtest` Test Parameter Options Dialog Box

TABLE 59-1 `vmemtest` Options

vmemtest Options	Description
Mode	Two modes are available: • Regular mode tests the specified amount of memory as one block and passes as the size argument 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	Specifies the amount of memory to reserve from being tested by `vmemtest`. The test ensures that 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, this 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. Record the memory configuration details on the target system before selecting nondefault setting for the amount option. If negative values are specified, the test assumes the default setting. 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. This number ndicates the amount of virtual memory found, rounded up to the nearest Kbyte.
Contiguous Errors	Specifies the max number of contiguous memory errors, which will be considered and counted as one noncontiguous error. The default value is 10.
ECC Error monitoring	Enables or disables ECC error monitoring.
ECC Error threshold	Determines how many correctable ECC errors occurred in the elapsed time before `vmemtest` reports a test failure. The default threshold value is 2.

TABLE 59-1 `vmemtest` Options *(Continued)*

vmemtest Options	Description
Test Method	`vmemtest` runs the Sequential and File Caching subtests by default. • Sequential subtest – Tests the whole memory 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 – Noncontiguous 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. • Nonsequential striding – Tests randomly from first to last page, within a specified memory range. Alternates 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 59-2 vmemtest Supported Test Modes

Test Mode	Description
Functional	Writes a pattern to an amount of the specified virtual memory. The data is then read back and compared. If there is a miscompare, the virtual address is reported and the data is read again and compared.

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. On miscompare, the VA, corresponding PA, and the expected and observed data patterns are reported.

vmemtest Command-Line Syntax

/opt/SUNWvts/bin/vmemtest *standard_arguments* **-o mode=***type*, **reserve=** *n*, **amount=***n*, **cerr=***n*, **eccmon=***enable*|*disable*, **eccthreshold=***n*, **type1=** *enable*|*disable*, **pp1=***pattern*, **type2=***enable*|*disable*, **pp2=***pattern*, **type3=***n*, **pp3=** *pattern*, **type4=***enable*|*disable*, **pp4=***pattern*, **type5=***enable*|*disable*, **pp5=***pattern*, **type6=***enable*|*disable*, **up=***hex_pattern*

TABLE 59-3 vmemtest Command-Line Syntax

Argument	Explanation
mode=*Page*|*Regular*	Specifies which mode of the `vmemtest` to run. • `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.
eccmon=*enabled*|*disabled*	Enables or disables the ECC error monitor.

TABLE 59-3 `vmemtest` Command-Line Syntax

Argument	*(Continued)* Explanation
`eccthreshold=`*n*	Specifies how many correctable ECC errors can occur in the elapsed time before `vmemtest` reports a test failure.
`type1=`*value* **`pp1=`***pattern*	`type1` is the 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 the 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 the 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 the `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 the `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 the `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`.

APPENDIX **A**

Loopback Connectors

Loopback Connection Overview

Loopback connectors aid in testing communication ports. The connectors take the form of either a single plug or a port-to-port cable with some communication connections shorted (looped-back).

Note – Loopback connectors must be wired properly and connected firmly for the Serial Port tests to work correctly. Miswired, poorly soldered, or missing loopback connectors can cause erroneous diagnostic error messages.

The following table depicts the pin assignments for most loopback plugs and cables that may be used when testing a system.

TABLE A-1 Loopback Connector Pin Assignments

Signal Description	EIA	CCITT #	RS-449 A	RS-449 B	DIN 8 8-pin round	DB9 9-pin	DB25 25-pin	Direction	Alpha ID
Chassis/Frame Ground	AA	101	1	NC*	NC*	NC*	1	None	AA
Transmit Data (TxDa)	BA	103	4	22	3	3	2	Output	BA
Receive Data (RxDa)	BB	104	6	24	5	2	3	Input	BB
Request To Send (RTSa)	CA	105	7	25	6	7	4	Output	CA
Clear To Send (CTSa)	CB	106	9	27	2	8	5	Input	CB
Data Set Ready (DSRa)	CC	107	11	29	NC*	6	6	Input/ output	CC
Signal Ground (SG)	AB	102	19	NC*	4	5	7	None	AB
Data Carrier Detect (DCDa)	CF	109	13	31	7	1	8	Input	CF
Transmit Clock In (TRxCa)	DB	114	5	23	NC*	NC*	15	Input	DB
Receive Clock In (RTxCa)	DD	115	8	26	8	NC*	17	Input	DD
Data Terminal Ready (DTRa)	CD	108	12	30	1	4	20	Output	CD
External Clock Out (TRxCa)	DA	113	17	35	NC*	NC*	24	Output	DA
Secondary Data Carrier Detect (DCDb)	SCF	122	NC*	NC*	NC*	NC*	12	Input	SCF
Secondary Clear to Send (CTSb)	SCB	121	NC*	NC*	NC*	NC*	13	Input	SCB

TABLE A-1 Loopback Connector Pin Assignments *(Continued)*

Signal Description	EIA	CCITT #	RS-449 A	RS-449 B	DIN 8 8-pin round	DB9 9-pin	DB25 25-pin	Direction	Alpha ID
Secondary Transmit Data (TxDb)	SBA	118	NC*	NC*	NC*	NC*	14	Output	SBA
Secondary Receive Data (RxDb)	SBB	119	NC*	NC*	NC*	NC*	16	Input	SBB
Secondary Request to Send (RTSb)	SCA	120	NC*	NC*	NC*	NC*	19	Output	SCA

*NC = No connection

25-Pin RS-232 Loopback Plug

FIGURE A-1 shows the RS-232 and RS-423 single-port loopback plug, which is a specially wired male DB-25 connector. The plug is connected to a serial port in the back of the system under test. The following table lists the pin connections for connecting the first plug to the second plug:

TABLE A-2 25-pin RS-232 Loopback Plug Wiring Plug Connections

First Plug	Second Plug
Pin 3	Pin 2
Pins 6 and 8	Pin 20
Pins 15 and 17	Pin 24

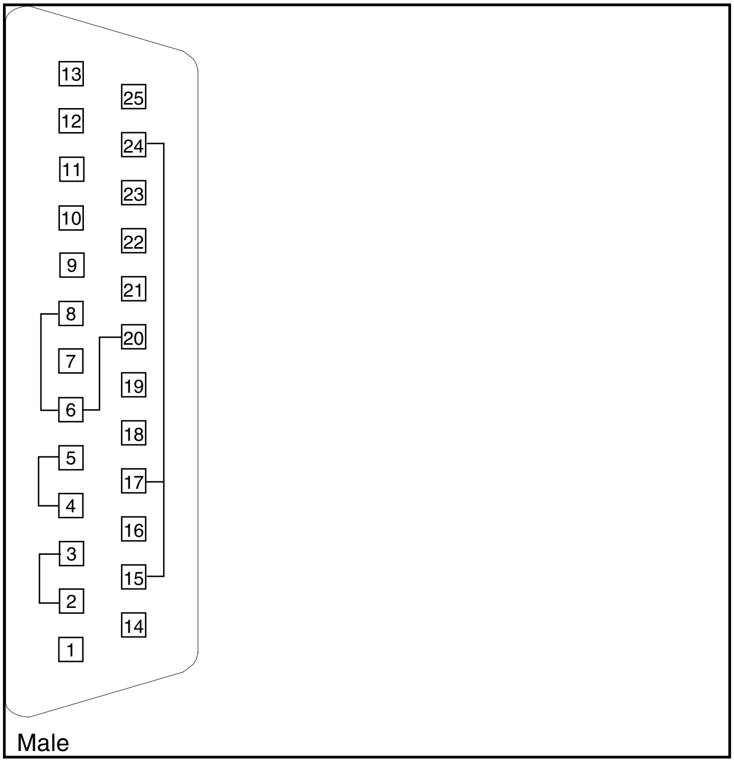


FIGURE A-1 25-pin RS-232 Loopback Plug Wiring Diagram

25-Pin RS-232 Port-to-Port Loopback Cable

FIGURE A-2 shows how to connect 25-pin RS-232 and RS-423 port to 25-pin RS 232 and RS 423 port loopback cables (two DB-25 connections). The cable is connected to a pair of serial ports in the back of the system under test. Both plugs are male. The following table lists the pin connections for connecting the first plug to the second plug:

TABLE A-3 25-pin RS-232 Port-to-Port Loopback Cable Wiring Plug Connections

First Plug	Second Plug
Pin 2	Pin 3
Pin 3	Pin 2
Pin 4	Pin 5
Pin 5	Pin 4
Pins 6 and 8	Pin 20
Pin 7	Pin 7
Pins 15 and 17	Pin 24
Pin 20	Pins 6 and 8
Pin 24	Pins 15 and 17

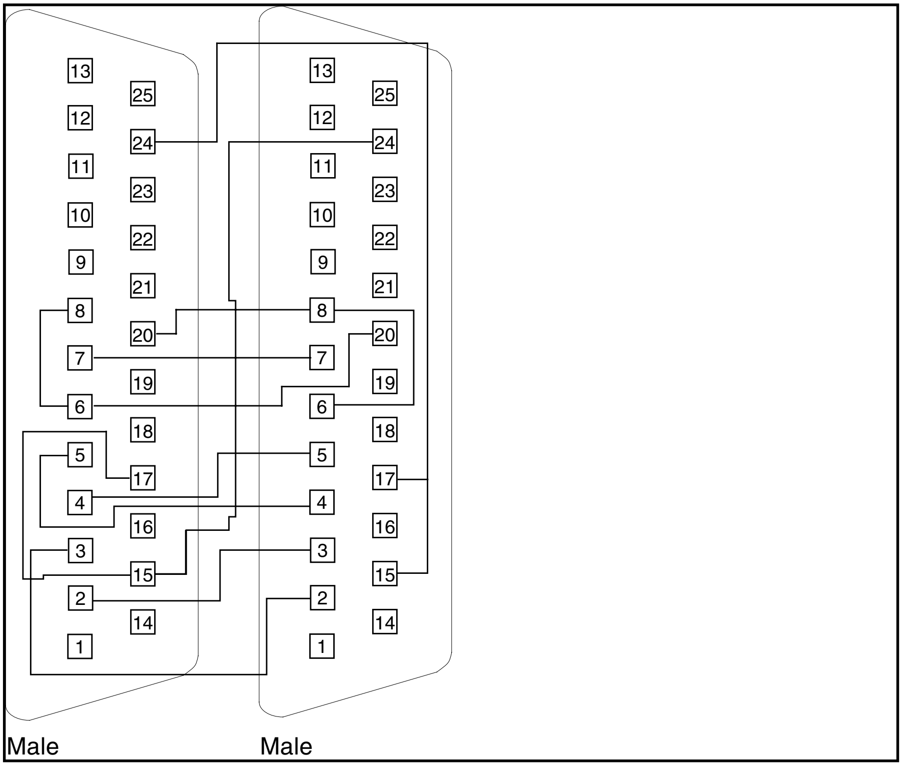


FIGURE A-2 25-pin RS-232 Port-to-Port Loopback Cable Wiring Diagram

8-Pin to 8-Pin Loopback Cable

FIGURE A-3 shows how to connect 8-pin round DIN RS-232 port to RS-423 to 8-pin round-DIN RS-232 and RS-423 port loopback cables. Both plugs are male. The following table lists the pin connections for connecting the first plug to the second plug:

TABLE A-4 8-Pin to 8-Pin Loopback Cable Wiring Plug Connections

First Plug	Second Plug
Pin 3	Pin 5
Pin 5	Pin 3

TABLE A-4 8-Pin to 8-Pin Loopback Cable Wiring Plug Connections *(Continued)*

First Plug	Second Plug
Pin 6	Pin 2
Pin 2	Pin 6
Pin 7	Pin 1

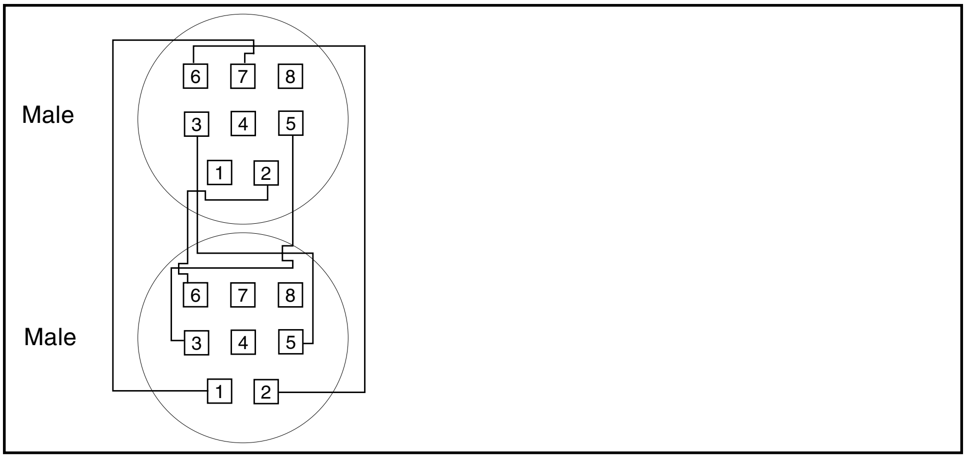


FIGURE A-3 8-Pin to 8-Pin Loopback Cable Wiring Diagram

Pin 8, Receive clock In (DD), remains unconnected.

8-Pin Loopback Plug

FIGURE A-4 shows how to connect male 8-pin round-DIN RS-232 and RS-423 single-port loopback plugs. The following table lists the pin connections for connecting the first plug to the second plug:

TABLE A-5 8-Pin Loopback Plug Wiring Plug Connections

First Plug	Second Plug
Pin 3	Pin 5
Pin 6	Pin 2
Pin 1	Pin 7

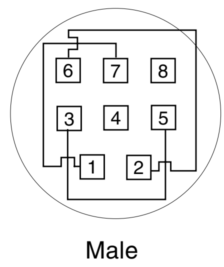


FIGURE A-4 8-Pin Loopback Plug Wiring Diagram

Pin 8, Receive Clock In (DD), remains unconnected.

25-Pin Port A-to-Port B Loopback Plug

FIGURE A-5 shows how to connect a 25-pin Port A to Port B loopback plug for most systems. The following table lists the pin connections for connecting the first plug to the second plug:

TABLE A-6 Port A-to-Port B Loopback Plug Wiring Plug Connections

First Plug	Second Plug
Pin 16	Pin 2
Pin 3	Pin 14
Pin 13	Pin 4
Pin 5	Pin 19
Pins 6 and 8	Pin 11
Pin 12	Pin 20
Pin 18	Pin 24
Pins 15 and 17	Pin 25

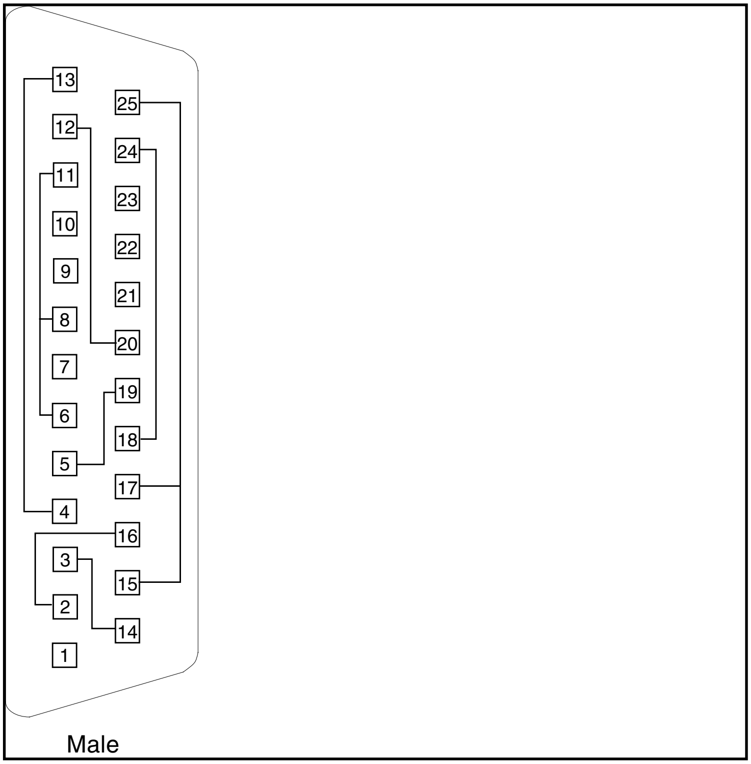


FIGURE A-5 Port A-to-Port B Loopback Plug Wiring Diagram

25-Pin Port A-to-A Port B-to-B Loopback Plug

If your system has a single communications port to connect it to peripherals, follow the connection instructions in FIGURE A-6 to make a male 25-pin loopback plug for that communication port. The following table lists the pin connections for connecting the first plug to the second plug:

TABLE A-7 Port A-to-A, Port B-to-B Loopback Plug Wiring Plug Connections

First Plug	Second Plug
Pin 3	Pin 2
Pin 5	Pin 4
Pins 6 and 8	Pin 20
Pin 12	Pin 11
Pin 13	Pin 19
Pin 16	Pin 14
Pins 15 and 17	Pin 24
Pin 25	Pin 18

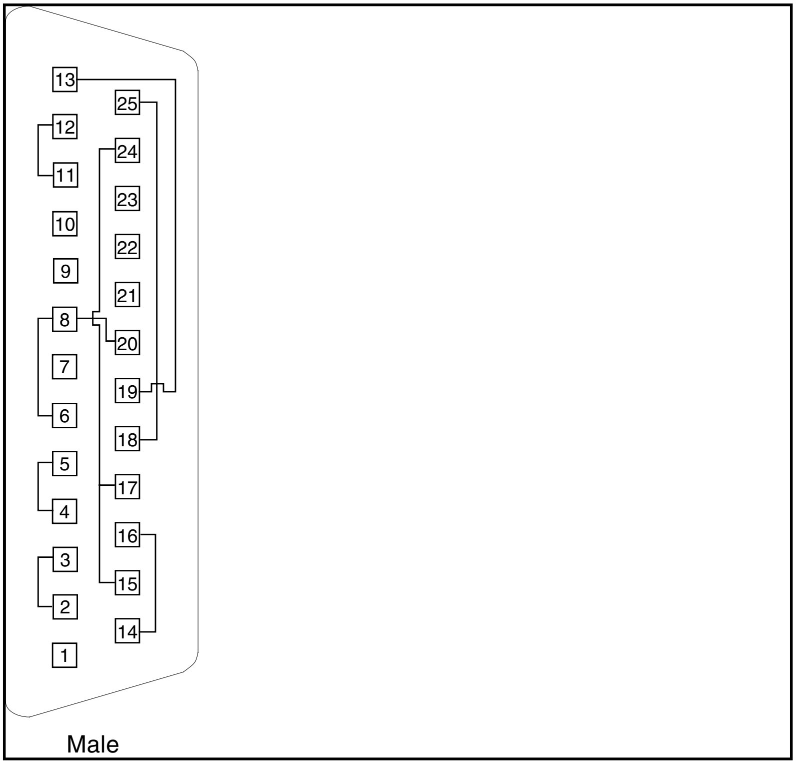


FIGURE A-6 Port A-to-A, Port B-to-B Loopback Plug Wiring Diagram

96-Pin Female Loopback Connector

FIGURE A-7 shows a 96-pin connector that can be ordered from Sun (part number 370-1366). The following table lists the pin connections for connecting the first plug to the second plug:

TABLE A-8 96-Pin Female Loopback Connector Wiring Plug Connections

First Plug	Second Plug
Pins 4 and 12	Pin 77
Pin 5	Pin 20
Pins 6	Pin 36
Pin 10	Pin 11

TABLE A-8 96-Pin Female Loopback Connector Wiring Plug Connections *(Continued)*

First Plug	Second Plug
Pin 13	Pin 16
Pin 14	Pin 15
Pin 18	Pin 19
Pin 21	Pin 24
Pin 28	Pin 60
Pin 29	Pin 68
Pin 30	Pin 34
Pin 37	Pin 40
Pin 38	Pin 39
Pin 42	Pin 43
Pin 45	Pin 48
Pin 46	Pin 47
Pin 52	Pin 78
Pin 53	Pin 55
Pin 54	Pin 75
Pin 58	Pin 59
Pin 61	Pin 64
Pin 62	Pin 63
Pin 66	Pin 67
Pin 69	Pin 72
Pin 76	Pin 79
Pin 82	Pin 83
Pin 85	Pin 88
Pin 86	Pin 87
Pin 90	Pin 91
Pin 93	Pin 96
Pin 94	Pin 95

The following are the materials used for this plug:

- PCR-E96FA(1)
- PCS-E96LKPA(1)
- 3751 Metal Plug(1) (9563K42)

- AWG28 Madison Cable(8” long) UL/CSA Approved

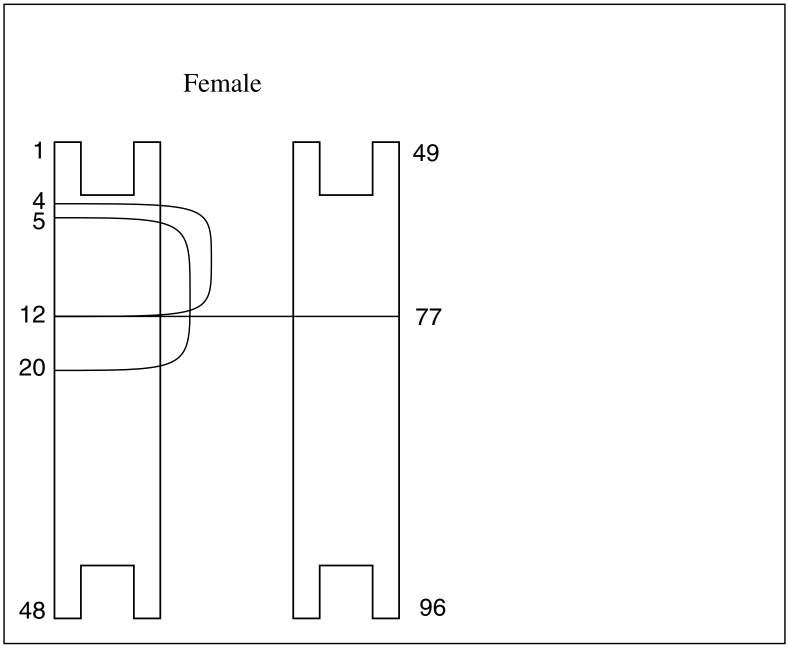
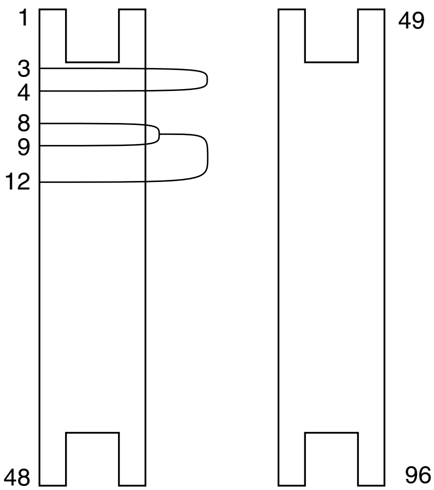


FIGURE A-7 96-Pin Female Loopback Connector Wiring Diagram

96-Pin Female Special Loopback Connector

This 96-pin connector can be ordered from Sun (part number 370-1381).

Female Special



Connect:

First Connector	to	Second Connector
pin 3		pin 4
pin 5		pin 7
pins 8 and 9		pin 12
pin 10		pin 11
pin13		pin 14
pin 15		pin 17
pins 18 and 19		pin 22
pin 20		pin21
pin 27		pin 28
pin 29		pin 31
pins 32 and 33		pin 36
pin 34		pin 35
pin 37		pin 38
pin 39		pin 41
pins 42 and 43		pin 46
pin 44		pin 45
pin 51		pin 52
pin 53		pin 55
pins 56 and 57		pin 60
pin 58		pin 59
pin 61		pin 62
pin 63		pin 65
pins 66 and 67		pin 70
pin 68		pin 69
pin 75		pin 76
pin 77		pin 79
pins 80 and 81		pin 84
pin 82		pin 83
pin 85		pin 86
pin 87		pin 89
pins 90 and 91		pin 94
pin 92		pin 93

Materials:
PCR-E96FA(1)
PCS-E96LKPA(1)
9563K999 Cap(1)
Madison Cable(6” long)
(#28 SCSI UL/CSA Approved)

Open Contacts:
1, 2, 6,16, 23, 24, 25, 26, 30, 40,
47, 48, 49, 50, 54, 64, 71, 72, 73,
74, 78, 88, 95, 96

FIGURE A-8 96-Pin Female Special Loopback Connector Wiring Diagram

37-Pin RS-449 Loopback Cable

Use these wiring instructions for a loopback cable for two 37-pin RS-449 synchronous ports.

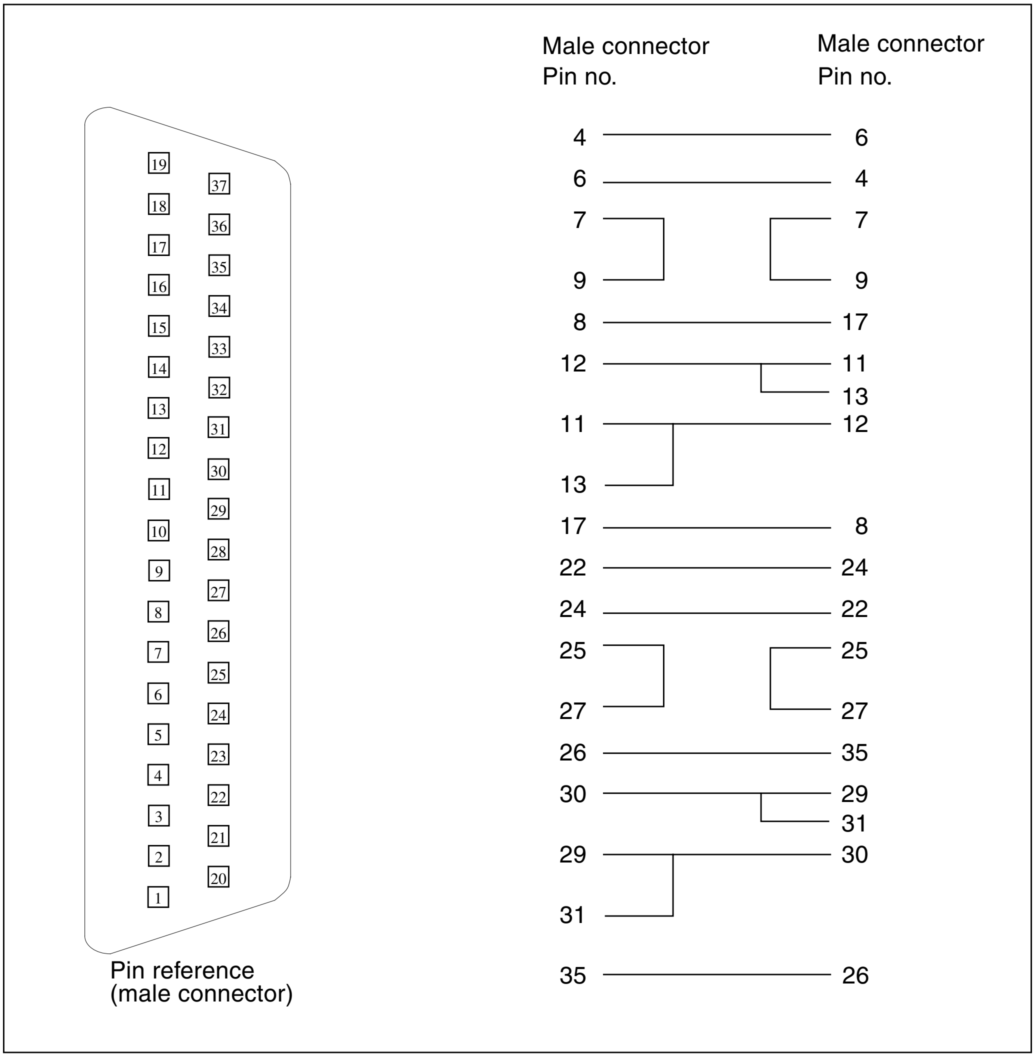


FIGURE A-9 37-Pin RS-449 Loopback Cable Wiring Diagram

37-Pin RS-449 Loopback Plug

Use these wiring instructions to make a male 37-pin RS-449 loopback plug. This plug is also available from Sun (part number 530-1430).

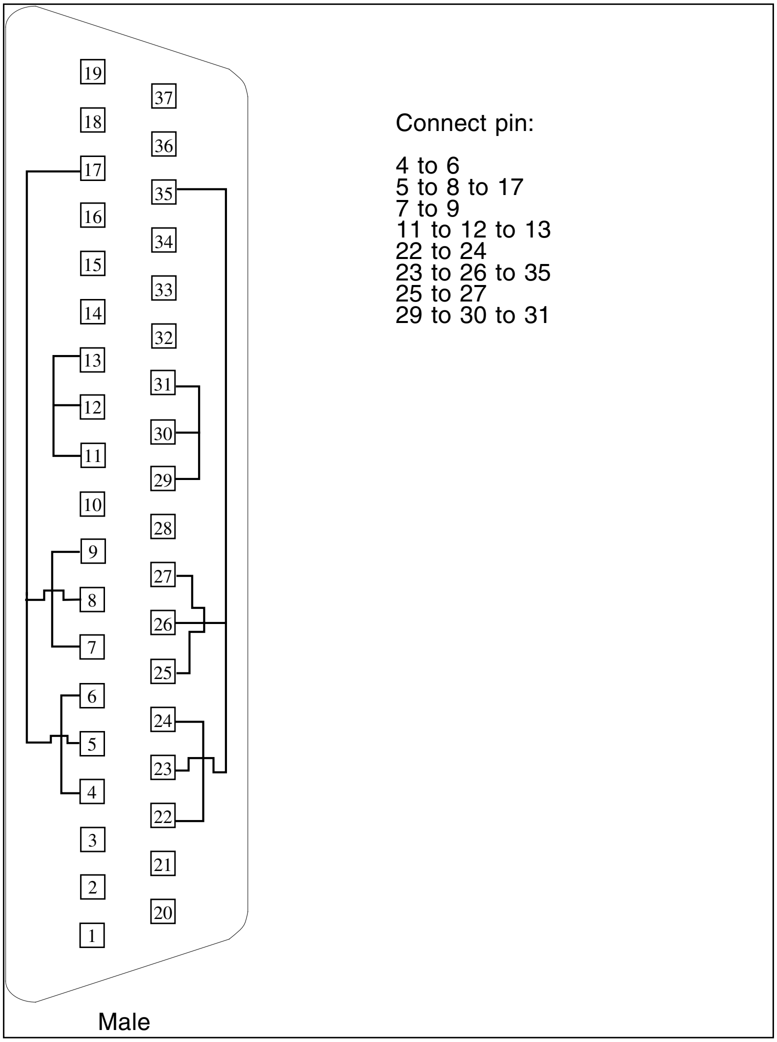
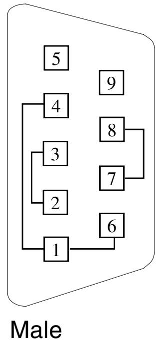


FIGURE A-10 37-Pin RS-449 Loopback Plug Wiring Diagram

9-Pin Male Single-Port Loopback Plug

Use these wiring instructions for male 9-pin RS-232 and RS-423 single-port loopback plugs.



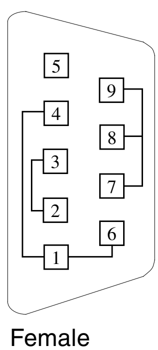
Connect:

First connector	to	Second connector
Pin 2		Pin 3
Pin 1		Pins 4 and 6
Pin 7		Pin 8

FIGURE A-11 9-Pin Male Single-Port Loopback Plug Wiring Diagram

9-Pin Female Single-Port Loopback Plug

Use these wiring directions for female 9-pin RS-232 and RS-423 single-port loopback plugs. Use this loopback plug with the `pcmciatest`.



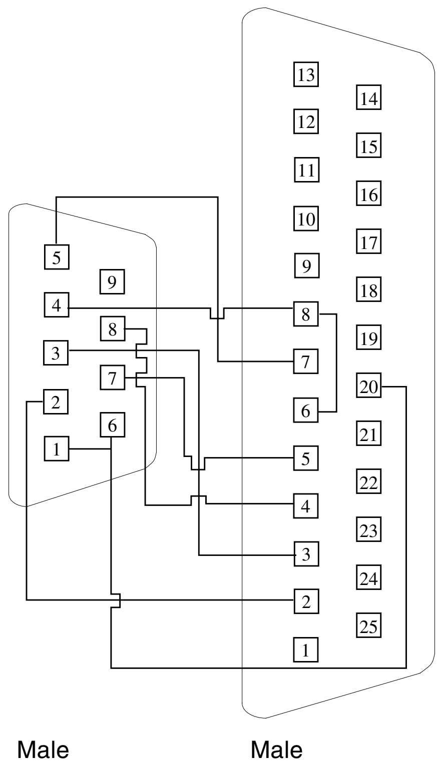
Connect:

First connector	to	Second connector
Pin 2		Pin 3
Pin 1		Pins 4 and 6
Pin 7		Pins 8 and 9

FIGURE A-12 9-Pin Female Single-Port Loopback Plug Wiring Diagram

9-Pin to 25-Pin Port-to-Port Loopback Cable

Use these wiring instructions for a 9-pin RS-232 and RS-423 port to 25-pin RS-232 and RS 423 port loopback cables. Both connectors are male.



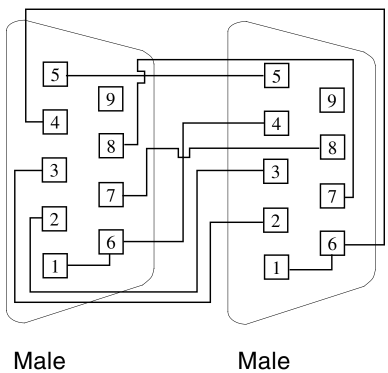
Connect:

First connector	to	Second connector
Pin 2		Pin 2
Pin 3		Pin 3
Pin 8		Pin 4
Pin 7		Pin 5
Pin 5		Pin 7
Pin 4		Pins 6 and 8
Pins 1 and 6		Pin 20

FIGURE A-13 9-Pin to 25-Pin Port-to-Port Loopback Cable Wiring Diagram

9-Pin to 9-Pin Port-to-Port Loopback Cable

Use these wiring instructions for 9-pin RS-232 and RS 423 port to 9-pin RS-232 and RS-423 port loopback cables. Both plugs are male.



Connect:

First connector	to	Second connector
Pins 1 and 6		Pin 4
Pin 2		Pin 3
Pin 3		Pin 2
Pin 4		Pins 1 and 6
Pin 5		Pin 5
Pin 7		Pin 8
Pin 8		Pin 7

FIGURE A-14 9-Pin to 9-Pin Port-to-Port Loopback Cable Wiring Diagram

This cable has no Sun part number assigned to it.

NT to TE Loopback Cable

Using two standard RJ-45 connectors, and connect pin 1 to pin 1, pin 2 to pin 2, and so on, for all pins. This loopback is a straight-through connection.

Twisted-Pair Ethernet (TPE) Loopback Cable for Fast Ethernet

Use the following wiring instructions for standard RJ-45 connectors for Fast Ethernet. Use loopback cable in `netlbtest` for `eri` devices.

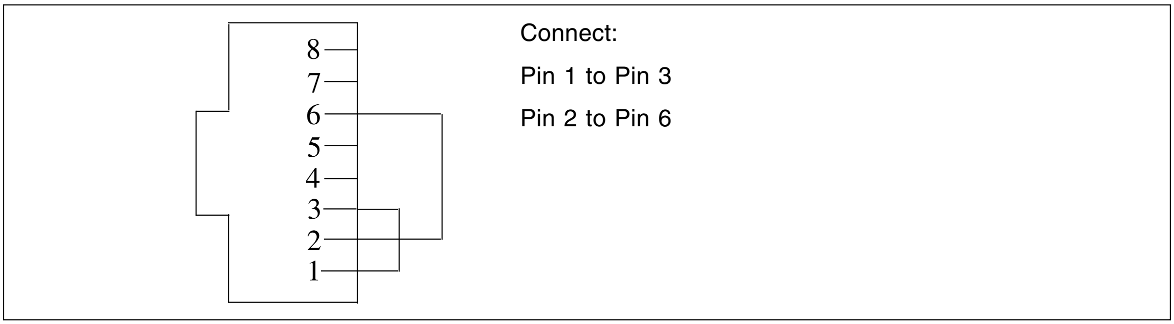
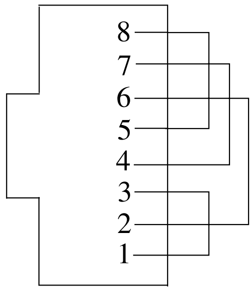


FIGURE A-15 Twisted-Pair Ethernet (TPE) Loopback Cable Wiring Diagram for Fast Ethernet

TPE Loopback Cable for Gigabit and 10/100 Ethernet

Use the following wiring instructions for RJ-45 plugs for Gigabit and 10/100 Ethernet. This loopback cable is used in `netlbtest` for Gigabit and 10/100 devices.



Connect:

Pin1 to Pin 3

Pin 2 to Pin 6

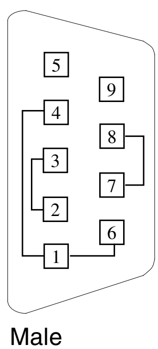
Pin 4 to Pin 7

Pin 5 to Pin 8

FIGURE A-16 TPE Loopback Cable Wiring Diagram for Gigabit and 10/100 Ethernet

9-Pin Male Single-Port Loopback Plug

Use these wiring instructions for male 9-pin RS-232 and RS-423 single-port loopback plugs.



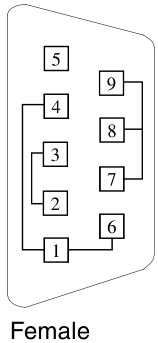
Connect:

First connector	to	Second connector
Pin 2		Pin 3
Pin 1		Pins 4 and 6
Pin 7		Pin 8

FIGURE A-17 9-Pin Male Single-Port Loopback Plug Wiring Diagram

9-Pin Female Single-Port Loopback Plug

Use these wiring directions for female 9-pin RS-232 and RS-423 single-port loopback plugs. Use this loopback plug with the `pcmciatest`.



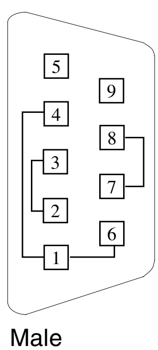
Connect:

First connector	to	Second connector
Pin 2		Pin 3
Pin 1		Pins 4 and 6
Pin 7		Pins 8 and 9

FIGURE A-18 9-Pin Female Single-Port Loopback Plug Wiring Diagram

9-Pin Male DB-9 External Loopback Connector

Use these wiring instructions for male 9-pin DB-9 external loopback plugs.



Connect:

First connector	to	Second connector
Pin 2		Pin 3
Pin 1		Pins 4 and 6
Pin 7		Pin 8

FIGURE A-19 9-Pin Male DB-9 External Loopback Connector Wiring Diagram

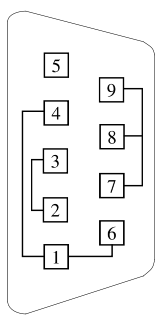
The signals and names for the DB-9 connector are as follows.

Pin	Name	Signal
1	DCD	Data Carrier Detect
2	RxD	Receive Data
3	TxD	Transmit Data

Pin	Name	Signal
4	DTR	Data Terminal Ready
5	SGND	Signal Ground
6	DSR	Data Set Ready
7	RTS	Request To Send
8	CTS	Clear To Send
9	RI	Ring Indicator

9-Pin Female DB-9 External Loopback Connector

Use these wiring instructions for female 9-pin DB-9 external loopback connectors.



Female

Connect:

First connector	to	Second connector
Pin 2		Pin 3
Pin 1		Pins 4 and 6
Pin 7		Pins 8 and 9

FIGURE A-20 9-Pin Female DB-9 External Loopback Connector Wiring Diagram

Glossary

administrative domain An arbitrary collection of hosts and networks that are monitored by the software as a single hierarchal entity. You may choose to divide their enterprise into several domains, each to be managed by different users.

agent A software process, usually corresponding to a particular local managed host, that carries out manager requests and makes local system and application information available to remote users.

bus A point-to-point network component. Used by the software to represent a network link to which many other hosts may be connected.

community A string similar to a password that is used to authenticate access to an agent's monitored data.

COMA Cache Only Memory Architecture. In a COMA machine, additional hardware including tag and state memory is added to the DRAM of each processing node to convert it into a kind of cache called attraction memory. See also NUMA.

CTQ Critical to Quality. A Sun Sigma terminology, used to identify key issues for a product based on voice of the customer.

diagnosis Correct interpretation and reporting of error.

diagnosability Ability of system to detect and correctly report errors when they occur.

diagnostics A test to uncover faults. These tests are used in design and validation to find margins and set limits on error protection, and stress tests used in operations to screen finished product for escapes.

diagnostic harness The harness or environment that schedules, manages, and controls execution of diagnostic tests on a platform.

DRAM Dynamic Read Access Memory. A type of semiconductor random access memory that stores information in integrated circuits that contain capacitors. Because capacitors lose their charge over time, the dynamic RAM must be periodically "refreshed" or recharged. Contrast with SRAM. See also VRAM.

EEPROM Electronically Erasable Programmable Read Only Memory. A special type of PROM that can be erased by exposing it to an electrical charge. Like other types of PROM, EEPROM retains its contents even when the power is turned off. Also like other types or ROM, EEPROM is not as fast as RAM. See also PROM, ROM, and RAM.

FC-AL Fibre Channel-Arbitrated Loop A connector that provides high bandwidth, increased distance, and additional connectivity from host to peripherals.

FIFO First-In First-Out Memory that stores data in queue order so the first input element goes out the first.

FRU Field Replaceable Unit.

GUI The graphical user interface, or GUI, is a window that provides the user with a method of interacting with the computer and its special applications, usually with a mouse or other selection device. The window usually includes such things as windows, an intuitive method of manipulating directories and files, and icons.

HDLC High Level Data Link. A common layer two protocol that handles both data and control messages.

hop The number of routers a packet goes through before reaching its destination.

module A software component that may be loaded dynamically to monitor data resources of systems, applications and network devices.

node A node is a workstation or server.

OpenBoot Diagnostics Standalone operation, without native operating system. Interactive menu driven, which provides ability to run tests from OpenBoot PROM level.

offline diagnostics Diagnostics that require the aid of the native operating system to function. Primarily tests components not currently in use by customer.

online diagnostics Diagnostics that require the aid of the native operating system to function. Primarily tests components currently in use by customer.

NUMA Non-Uniform Memory Access. CC/NUMA or CC-NUMA is Cache-Coherent Non-Uniform Memory Access. In a CC-NUMA machine, the physical address on the memory bus of a processing node is used to determine the home node memory location of a particular datum. The cache hierarchy on each processing node is constructed to replicate and hold copies of data from not only the local memory, but also the memory of remote nodes.

PICL Platform Information and Control Library.

POST Power on Self Test. Standalone operation, without native operating system. Mostly non interactive, automated tests, run when power is applied to the system.

production environment One of two environments in which software is deployed. The production environment is a “real” environment (as opposed to a test environment) in which you manage and monitor your hardware.

PROM Programmable Read Only Memory A memory chip on which data can be written only once. When a program has been written onto a PROM, it remains there forever. Unlike RAM, PROMs retain their data even when the power is turned off. See also RAM.

RAM Random Access Memory A type of computer memory that can be accessed randomly; that is, any byte of memory can be accessed without touching the preceding bytes. RAM is the most common type of computer and device memory.

ROM Read Only Memory Computer memory on which data has been prerecorded. Once data has been written onto a ROM chip, it cannot be removed and can only be read.

RTS/CTS Ready to Send/Clear to Send A hardware flow control handshake protocol used with serial lines.

SEEPROM Serial EEPROM. See also EEPROM.

segment An object representing a “segment” of the network, and used as a basis for a local network.

SNMP Simple Network Management Protocol. A simple protocol designed to allow networked entities (hosts, routers, and so on) to exchange monitoring information.

SNMPv2 usec SNMP version 2, user-based security model security standards.

SRAM Static Random Access Memory. A faster and more reliable RAM than dynamic random access memory (DRAM). DRAM offers access times of about 60 nanoseconds, while SRAM access can be as low as 10 nanoseconds. It is static because it does not require refreshment as does DRAM. See also VRAM.

standalone diagnostics Diagnostics that run without the aid of the native operating system. Usually an operating environment is built to provide basic scheduling capabilities.

standard error An open file normally connected directly to a primary output device, such as a terminal, printer, or screen. Error messages and other diagnostic output normally go to this file and then to the output device. You can redirect the standard error output into another file instead of to the printer or screen.

standard input (Standard input device) The device from which a program or system normally takes its input. Usually a terminal or the keyboard.

standard output (Standard output device) The device to which a program or system normally sends its output. Usually a terminal or the screen.

URL Uniform Resource Locator. A URL is a textual specification describing a resource which is network-accessible.

VRAM Video Random Access Memory. A type of dynamic RAM (DRAM) used in high-speed graphics frame buffers. With conventional DRAM, both the processor and the frame buffer logic must access RAM by sharing the same signal lines or buses on the RAM chips. VRAM provides separate buses for the processor and the frame buffer logic. See also DRAM, SRAM.

Index

S

T

V

W