SunVTS™ 6.0 Patch Set 3 Documentation Supplement for x86 Platforms

Sun Microsystems, Inc.
www.sun.com

Part No. 819-3776-10
September 2005, Revision A

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

Copyright 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at http://www.sun.com/patents and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and in other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

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

Copyright 2005 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, Californie 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuels relatants à la technologie qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuels peuvent inclure un ou plus des brevets américains énumérés à http://www.sun.com/patents et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une license non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciées de Sun qui mettent en place l'interface d 'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

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

Adobe PostScript™

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), in the Java Desktop System (JDS) or the Common Desktop Environment (CDE). 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 supplement to the *SunVTS 6.0 Test Reference Manual* (819-3777-10) and the *SunVTS 6.0 User's Guide* (817-7664-10). Refer to these documents for details on the full collection of SunVTS 6.0 tests and software features that are not new or enhanced in this release.

This document describes SunVTS Version 6.0 Patch Set 3 tests that are distributed on the Solaris 10 HW2 Software DVD and CDs.

Note – The Solaris™ release with which this version of SunVTS is delivered supports systems that use the x86 families of processor architectures (AMD64, Pentium, and Xeon EM64T). The supported systems appear in the *Solaris 10 Hardware Compatibility List* at `http://www.sun.com/bigadmin/hcl`.

Before You Read This Book

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

- *SunVTS 6.0 User's Guide*

- *SunVTS 6.0 Test Reference Manual*
- *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 new or enhanced 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*`%`
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 6.0 Patch Set 3 tests. It serves as a reference companion to the SunVTS documents listed below.

Application	**Title**	**Part Number**
Test Parameter Reference	*SunVTS 6.0 Test Reference Manual*	817-7665-10
Installation and Navigation	*SunVTS 6.0 User's Guide*	817-7664-10
Quick Reference Card	*SunVTS Quick Reference Card*	817-7686-10

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 6.0 Patch Set 3 Documentation Supplement for x86 Platforms, part number 819-3776-10

CHAPTER 1

Introduction

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

This manual is a supplement to the SunVTS™ 6.0 documentation and describes new features, tests, and test enhancements that are developed in the SunVTS 6.0 patch releases. The items described in this document are provided in the SunVTS 6.0 Patch Set 3 software.

For overall SunVTS features, test configuration modes, interfaces, and options refer to the *SunVTS 6.0 User's Guide*. Refer to the *SunVTS 6.0 Test Reference Manual* for detailed information on SunVTS test software and the full collection of tests released with SunVTS 6.0.

Refer to the latest version of the *SunVTS 6.0 Patch Set 3 Release Notes* (819-2949-10) for important release information and open issues. This document is available at: `http://www.sun.com/documentation`

New Features

The Emulex HBA test (`emlxtest`) is new in this release.

New Test Library Function

The test library consists of a new function, `vts_escape_comma(char *)`. This function enables you to insert escape characters in front of commas in the configuration string.

Function name: `vts_escape_comma`
Input parameters: `char *`
Return Value: `char *`

The following is an example of how to call the function from `testnametest_info.c`:

```
In
----------------
vts_register
{
char *cfg_string, s1[1024];
strncpy(s1, "Test, Configuration, information",1024);
cfg_string = vts_escape_comma(s1);
 :
 :
TEST_CONFIG, cfg_string
  :
  :
}
if (cfg_string) free(cfg_string);
-------------------
```

x86 and 64-Bit Solaris Support

Starting with Solaris 10, the SunVTS infrastructure and core diagnostics are available for x86 Solaris platforms. Starting with Solaris 10 7/05, SunVTS x86 diagnostics are supported in the AMD 64-bit environment for the SunVTS kernel (`vtsk`). All x86 diagnostics except the System Test (`systest`) are ported to 64-bit.

SunVTS is supported and tested on the following Sun x86 platforms:

- Sun Fire V60x
- Sun LX50
- Sun Fire B100x
- Sun Fire B200x
- Sun Fire V20z
- Sun Fire V40z

You must install the x86 version of the SunVTS packages to perform SunVTS on x86 platforms. The software packages use the same names as in the SPARC environment. The SunVTS packages delivered separately for both x86 and SPARC Solaris platforms are as follows:

- `SUNWvts` – Contains the SunVTS core framework that includes the kernel and user interface.
- `SUNWvtsmn` – Contains the SunVTS online manual pages
- `SUNWvtsr` – Contains the SunVTS framework configuration files in the root partition (Superuser).
- `SUNWvtsts` – Contains the SunVTS test binaries.

The SunVTS components available for x86 Solaris platforms are as follows.

Infrastructure:

- `sunvts`
- `vtsk`
- `vts_cmd`
- `vtstty`
- `vtsui`
- `vtsprobe`

SunVTS Tests:

- CD DVD Test (`cddvdtest`)
- CPU Test (`cputest`)
- Disk and Floppy Drives Test (`disktest`)
- Data Translation Look-aside Buffer (`dtlbtest`)
- Floating Point Unit Test (`fputest`)
- InfiniBand Host Channel Adapter (`ibhcatest`)
- Level 1 Data Cache Test (`l1dcachetest`)
- Ethernet Loopback Test (`netlbtest`)
- Network Hardware Test (`nettest`)
- Physical Memory Test (`pmemtest`)
- RAM test (`ramtest`)
- Serial Port Test (`serialtest`)
- System Test (`systest`)
- Universal Serial Board Test (`usbtest`)
- Virtual Memory Test (`vmemtest`)

SunVTS Overview

SunVTS is composed of many individual tests that support testing of a wide range of products and peripherals. Most of the tests are capable of testing devices in a 32-bit or 64-bit Solaris environment.

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

- Audio tests

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

Such flexibility means that the proper test modes and options need to be selected to maximize its effectiveness.

Note – When an error occurs in VTS 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 manual.

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 6.0 User's Guide* for installation information.

Test Requirements

SunVTS 6.0 and 6.0 patch set releases are supported in the Solaris 10 Operating System and future compatible Solaris releases.

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 book.

Getting SunVTS Version Information

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

Collection of SunVTS Tests

Many individual tests make up the collection of tests in the SunVTS application. 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.

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

32-Bit and 64-Bit Tests

Because each test is a separate program, you can run individual tests directly from the command line. When this is done, care must be taken to run the appropriate test (32-bit or 64-bit) that corresponds to the operating system that is running (32-bit or 64-bit). This is done by running tests from specific directories as follows:

- 32-bit tests—`/opt/SUNWvts/bin/`*testname*
- 64-bit tests—`/opt/SUNWvts/bin/64/`*testname*
    - The test is an actual 64-bit binary test if *testname* is a binary file.
    - The test is a 32-bit test capable of running in the 64-bit environment if *testname* is a symbolic link.

Note – The `SUNWvtsx` package must be installed for 64-bit SunVTS support. For more information on SunVTS packages and installation procedures refer to the *SunVTS 6.0 User's Guide*.

If you use the `sunvts` command to run SunVTS, SunVTS automatically allocates 32-bit or 64-bit tests based on the 32-bit or 64-bit Solaris operating environment that is running. Therefore, the only time that you need to be concerned with the 32-bit or 64-bit operation is when you run the SunVTS kernel or SunVTS tests from the command line.

If you are not sure which operating system is running, refer to the Solaris System Administration manuals. In Solaris 8 2/02 and Solaris 9, the following command can be used 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 various interfaces: The JDS or CDE graphical user interfaces, 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 9). 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 interfaces (GUIs)	Users can select tests and test options by pointing and clicking with a mouse button in the JDS or CDE GUI.
TTY interface	Users can run SunVTS from a terminal or modem attached to a serial port. This feature requires that users use the keyboard instead of the mouse, and it displays one screen of information at a time.
Command-line execution	Users can run each of the SunVTS tests individually from a shell tool command line using the command-line syntax. Each test description in this book contains the corresponding command-line syntax.

Note – To increase or decrease a numeric value in a SunVTS JDS or CDE dialog box, you can use either the up or down arrows, or type a new value in the text box and press Return. Select Apply to apply all dialog box changes.

Running a Test From a User Interface

The common way to run SunVTS testing is through a SunVTS user interface—JDS or CDE, 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 6.0 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 lower set of buttons are generic and are described below.

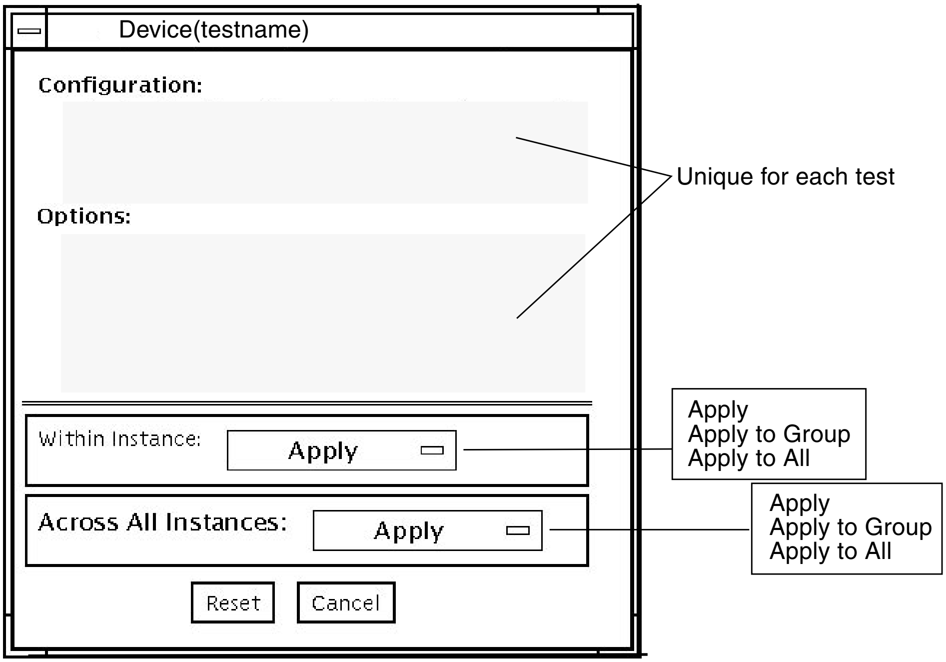


FIGURE 1-1 Test Parameter Options Dialog Box (JDS or CDE)

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, or • to all devices within this group with Apply to Group, or • 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, or • to all devices within this group with Apply to Group, or • 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 menu.
Cancel	Ignores any changes made to option values and closes the test parameter option menu.

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*. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the *SunVTS 6.0 Test Reference Manual*.

Standard Command-Line Arguments

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

TABLE 1-3 Standard Command-Line Arguments

Argument	Description
`-x`	Runs the test in exclusive test mode. This mode assumes that the test has exclusive access to the device and the system. The testing done in exclusive mode is usually more stressful compared to functional mode. Also, running a test in exclusive mode usually assumes exclusive access to all resources and assumes no other SunVTS test is running at the same time.
`-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.
`-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 test Trace mode and displays messages that track function calls and sequences currently in use by the test code. The default is false.
`-e`	Runs the test in Stress mode by increasing the system load. 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.
`-n`	Runs the test in Connection mode. See the note below. The default is false.
`-f`	Runs the test in full Functional test mode. This mode assumes that the test has complete control of the device under test. See the note below. The default is false.

TABLE 1-3 Standard Command-Line Arguments *(Continued)*

Argument	Description
`-i` *number*	Defines the number of instances for scalable tests.
`-w` *number*	Defines to which instance the test is assigned; this option is for scalable tests.
`-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

There are test-specific arguments, as described in TABLE 1-4. Test-specific arguments follow the format specified in the `getsubopt(3c)` man page. For information about test-specific arguments refer to the specific test chapter in this book.

TABLE 1-4 SunVTS Test-Specific Arguments

Argument	Description
`-o`	Separate each test-specific argument by commas, with no space after the comma. For example: **`#./sample -v -o dev=/dev/audio,volume=78`** The test option format is specified by the man page `getsubopt`(3C).

SunPCi-3 Card Support

`sunpci2test` now supports the SunPCi-3 cards. Solaris 10 supports SunPCi-3 Version 3.2.2 with Patch 118591-01 only. Solaris 10 does not support the SunPCi-2 card.

Testing Frame Buffers

Before running a frame buffer test, determine whether the test requires frame buffer locking. Not all frame buffer tests have a locking option. Some tests set the lock automatically. Check the test chapter for each individual test to see if this step is needed. If locking is required, you can set the lock in one of two ways:

- If you are using the JDS or CDE SunVTS interface, go to the Option menu of the graphic test and select Enable for the frame buffer locking option.
- If you are working from the command line, you can enable frame buffer locking with the `lock=e/d` option. For example, to run the generic frame buffer test (`fbtest`) with a locked frame buffer, enter:

```
# ./fbtest -o dev=pfb0,lock=enable
```

(See the test command line argument descriptions in the individual test chapters.)

Caution – If frame buffer locking is disabled (unlocked) on frame buffers that are running `vtsui`, or if you move the mouse, you will receive false error messages. Even a slight mouse movement can cause a test to fail.

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 User Collection. This document is available at: `docs.sun.com`.

Caution – If you are using the GUI 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 CDE or GNOME). 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.

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.

CHAPTER 2

BMC Environment Test (`bmcenvironment`)

The `bmcenvironment` test scans the sensors of the Base Management Controller (BMC) and reports failures on sensors that report readings outside of their reported ranges.

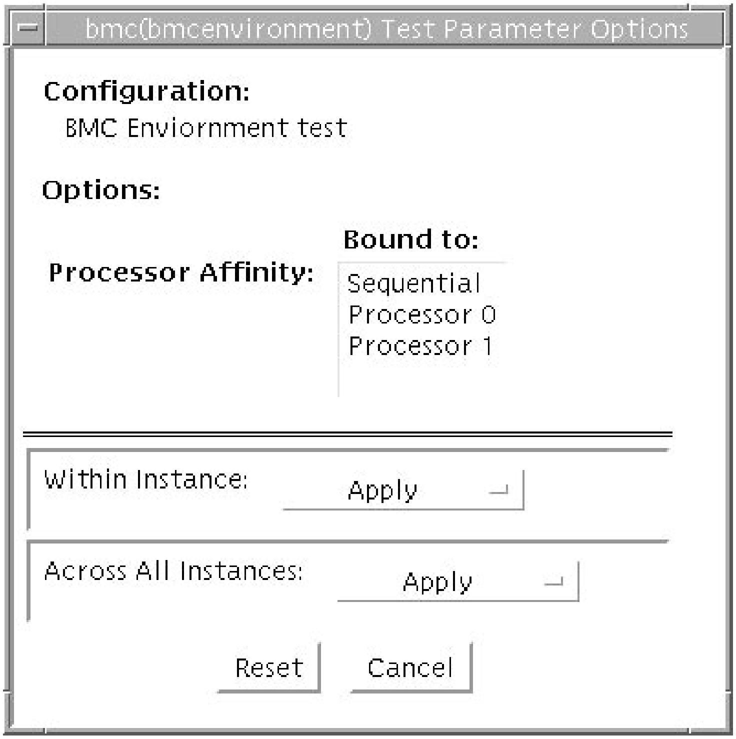
Note – The `bmcenvironment` test is supported on x86 platforms only.

Caution – Ensure that no other applications, such as `ipmitool`, are using the BMC when perfoming the `bmcenvironment` test.

Caution – Ensure that a service processor daughter card is installed and seated correctly.

`bmcenvironment` 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.



bmc(bmcenvironment) Test Parameter Options

Configuration:
BMC Enviornment test

Options:

Processor Affinity: **Bound to:**
Sequential
Processor 0
Processor 1

Within Instance: Apply

Across All Instances: Apply

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

Note – Only the standard options are supported for the `bmcenvironment` test.

`bmcenvironment` Test Modes

`bmcenvironment` supports the test modes listed in the following table.

TABLE 2-1 `bmcenvironment` Supported Test Modes

Test Mode	Description
Connection	Runs the entire test.

TABLE 2-1 `bmcenvironment` Supported Test Modes

Test Mode	Description
Functional	Runs the entire test.
Online	Runs the entire test.
Exclusive	Runs the entire test.

`bmcenvironment` Command-Line Syntax

`/opt/SUNWvts/bin/bmcenvironment` *standard_arguments*

Note – Only the standard SunVTS command line argumentst are supported for the `bmcenvironment` test.

CHAPTER **3**

Optical Disk Drive Test (`cddvdtest`)

`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 the CD-ROM, DVD-ROM 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 either blank or contain the SunVTS test data. When testing the write-once media, the media (such as CD-R) has to be blank at the start to run the write test. Such media could still run multiple passes of the test because after the first write test, the subsequent invocations will 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-ROM, 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

Note – `cddvdtest` is not a scalable test.

`cddvdtest` Hardware and Software Requirements

`cddvdtest` has different set of test requirements based on the media type as described below.

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 the SunVTS kernel associates the correct test options based on the media that is loaded in the drive.

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 test.

In case of non-blank media OR the media with non-SunVTS test data, the media must be blanked first by using the `cdrw` utility. Because -R or +R media can be written only once, only a 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 invocations will treat the media as Read Only and perform the test accordingly.

To prevent media corruption, if stopped during a write, finalize, format, or erase, the test posts a Warning message 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. It is better to set a limited number of passes for `cddvdtest`, instead of setting Max Passes=0 (unlimited) and stop the test manually. If a media is damaged, you should 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 is intended to make the test run less passes in long hours testing to preserve the media. It is also intended to give plenty of time to stop the test between passes.

`cddvdtest` Subtests

`cddvdtest` has different subtests based on the media type as described below.

CD-RW and DVD-RW

TABLE 3-1 `cddvdtest` Subtests for CD-RW and DVD-RW Media

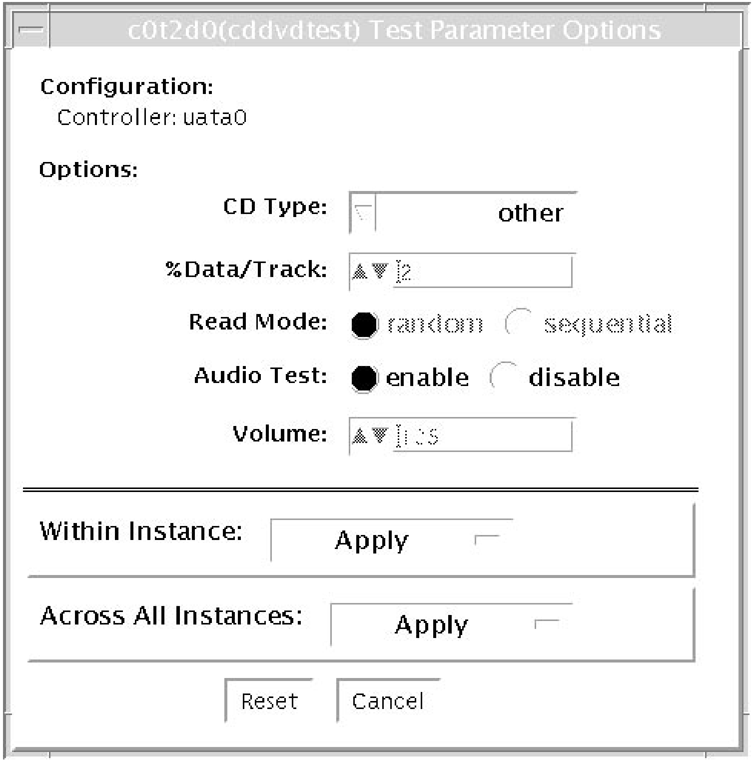
Subtest	Description
Blank	Erases data on DVD-RW media and CD-RW media if not blank. This subtest does not apply to DVD+RW media because this media type can not be blanked.
Simulation	Performs a write track with Laser turned off. This mode tests writing function without data written to the media.
Write	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 Content (TOC).
Eject	Ejects media.

`cddvdtest` Options

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

`cddvdtest` has different test options based on the media type as described below.

CD-ROM Test Options



c0t2d0(cddvdtest) Test Parameter Options

Configuration:

Controller: uata0

Options:

CD Type: other

%Data/Track: 2

Read Mode: ● random ○ sequential

Audio Test: ● enable ○ disable

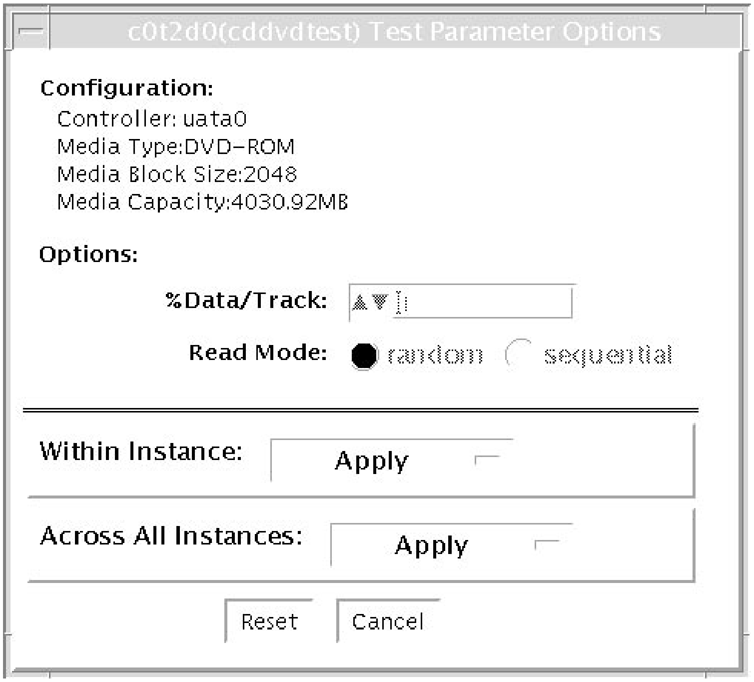
Volume: 125

FIGURE 3-1 `cddvdtest` Test Parameter Options Dialog Box for CD-ROM

TABLE 3-2 `cddvdtest` Options for CD-ROM

Option	Description
CD Type	The types of compact discs that can be tested are listed in the CD Type menu. The choices are: pdo, multi-session, or other the default CD type is other). In the 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 canned 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 canned 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



Configuration:

Controller: uata0
Media Type:DVD-ROM
Media Block Size:2048
Media Capacity:4030.92MB

Options:

%Data/Track:

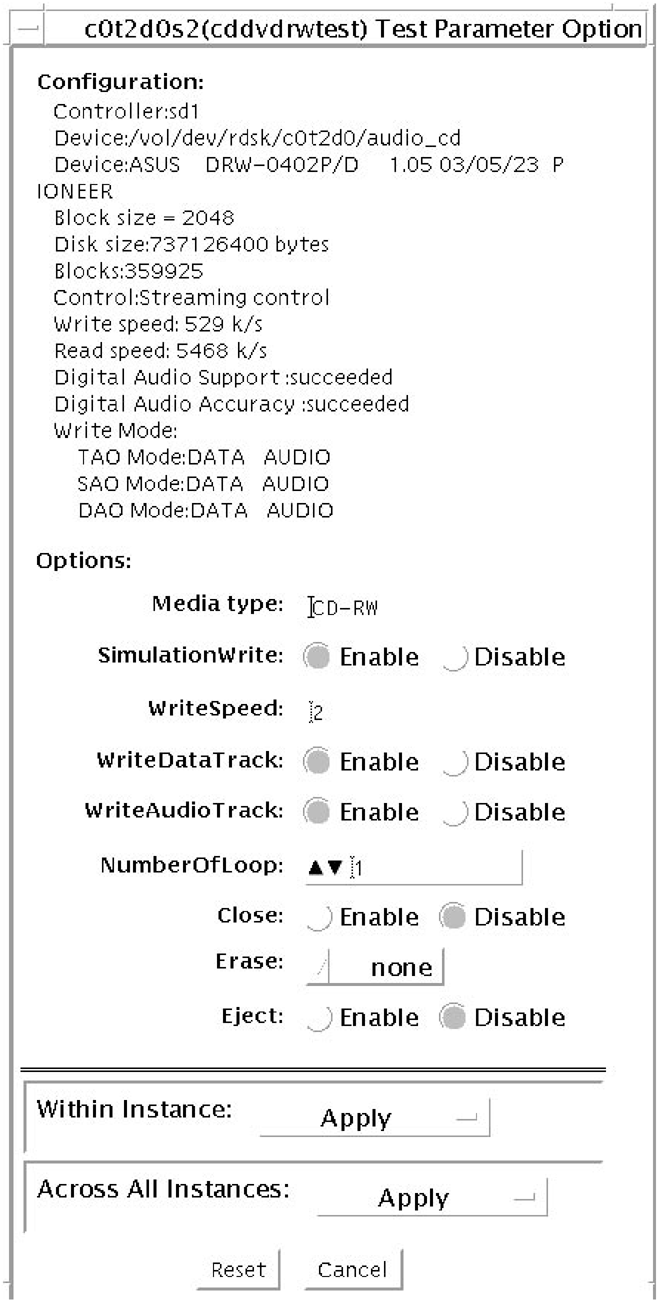
Read Mode: ● random ○ sequential

FIGURE 3-2 `cddvdtest` Test Parameter Dialog Box for DVD-ROM

TABLE 3-3 `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



c0t2d0s2(cddvdrwtest) Test Parameter Option

```
Configuration:
  Controller:sd1
  Device:/vol/dev/rdsk/c0t2d0/audio_cd
  Device:ASUS   DRW-0402P/D    1.05 03/05/23  P
IONEER
  Block size = 2048
  Disk size:737126400 bytes
  Blocks:359925
  Control:Streaming control
  Write speed: 529 k/s
  Read speed: 5468 k/s
  Digital Audio Support :succeeded
  Digital Audio Accuracy :succeeded
  Write Mode:
     TAO Mode:DATA   AUDIO
     SAO Mode:DATA   AUDIO
     DAO Mode:DATA   AUDIO
```

Options:

Media type: CD-RW

SimulationWrite: ● Enable ○ Disable

WriteSpeed: 2

WriteDataTrack: ● Enable ○ Disable

WriteAudioTrack: ● Enable ○ Disable

NumberOfLoop: 1

Close: ○ Enable ● Disable

Erase: none

Eject: ○ Enable ● Disable

Within Instance: Apply

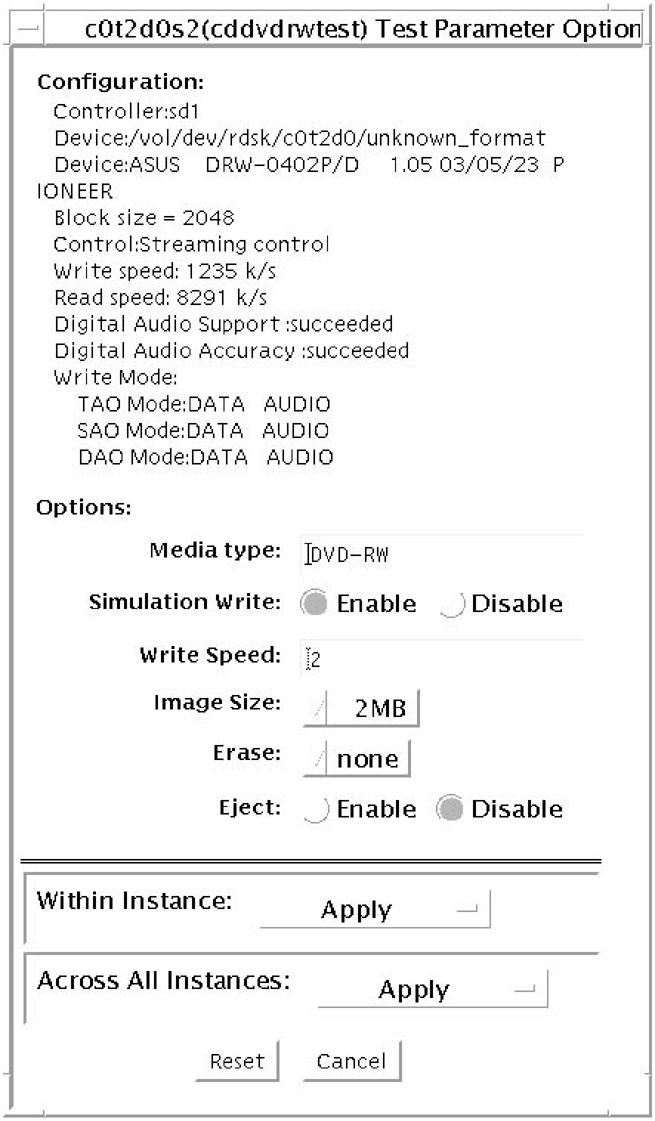
Across All Instances: Apply

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

TABLE 3-4 cddvdtest Options for CD-RW

Option	Description
SimulationWrite	Disable/Enable Simulation Write.
WriteSpeed	Speed entered in terms of *nX*. Speed will be set to closest approximation of *nX* allowed by device.
WriteDataTrack	Write a Data Track in one Test Loop.
WriteAudioTrack	Write an Audio Track in one Test Loop.
NumberOfLoop	Number of loop in one test pass comprising Data and Audio Track if enabled.
Close	Close after writing. Can not write more track on media.
Erase None	No erase All: Erase whole disk.
Eject	Eject the disk when test completed. Do not enable this option in a multiple pass test session.

DVD-RW Test Options



c0t2d0s2(cddvdrwtest) Test Parameter Option

Configuration:

```
  Controller:sd1
  Device:/vol/dev/rdsk/c0t2d0/unknown_format
  Device:ASUS    DRW-0402P/D    1.05 03/05/23  P
IONEER
  Block size = 2048
  Control:Streaming control
  Write speed: 1235 k/s
  Read speed: 8291 k/s
  Digital Audio Support :succeeded
  Digital Audio Accuracy :succeeded
  Write Mode:
     TAO Mode:DATA   AUDIO
     SAO Mode:DATA   AUDIO
     DAO Mode:DATA   AUDIO
```

Options:

Media type: DVD-RW

Simulation Write: ● Enable ○ Disable

Write Speed: 2

Image Size: 2MB

Erase: none

Eject: ○ Enable ● Disable

FIGURE 3-4 `cddvdtest` Test Parameter Options Dialog Box for DVD-RW

TABLE 3-5 `cddvdtest` Options for DVD-RW

Options	Description
SimulationWrite	Disable/Enable Simulation Write.
WriteSpeed	Speed entered in term of *nX*. Speed will be set to closest approximation of *nX* allowed by device.
ImageSize	Track Size, how much data is written: 2MB, 10MB, 2GB, 4GB.
Erase None	No erase.
Fast	Erase with "fast" option. Only Table of Content is erased.
All	Erase with "all" option. Erase whole disk surface.
Eject	Eject the disk when test completed. Do not enable this option in a multiple pass test session.

`cddvdtest` Supported 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 below.

CD-ROM Test Modes

TABLE 3-6 Supportted Test Modes for CD-ROM

Test Mode	Description
Connection	In this mode, `cddvdtest` verifies that a CD-ROM drive is connected to and configured in the system.
Functional	In this mode, 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 3-7 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 3-8 Supported Test Modes for CD-RW and DVD-RW

Test Mode	Description
Connection	Shows basic drive information and supporting modes. Shows disk Table of Content (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 Content. • Eject (optional).

`cddvdtest` Command-Line Syntax

`cddvdtest` has different command line syntax based on the media type as described below.

CD-ROM Command-Line Syntax

`/opt/SUNWvts/bin/sparcv9/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 3-9 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.
`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/sparcv9/cddvdtest` *standard_arguments* `-o dev=` *device_name*, `read=random|sequential`, `data=`*%_of_data*

TABLE 3-10 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/sparcv9/cddvdtest` *standard_arguments* `-o dev=`*cntndnsn*, `media=CD-RW`, `nosim`, `speed=`*n*, `nodata`, `noaudio`, `loop=`*n*, `close`, `erase={none, all}`, `eject`

TABLE 3-11 CD-RW Command-Line Syntax

Argument	Description
`dev=`*cntndnsn*	Specifies the device under test.
`media=CD-RW`	Specifies media.
`nosim`	Disables Simulation Write
`speed=`*n*	Specifies the speed entered in terms of *nX*.
`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` - Do not erase media after test complete. `all` - Erase entire disk.
`eject`	Ejects disk after test completed.

DVD-RW Command-Line Syntax

`/opt/SUNWvts/bin/sparcv9/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 3-12 DVD-RW Command-Line Syntax

Argument	Description
`dev=`*cntndnsn*	Specifies the devise under test.
`media={DVD-RW, DVD+RW}`	Specifies media.
`nosim`	.Disables Simulation Write
`speed=`*n*	.Specifies the speed entered in terms of *nX*.

TABLE 3-12 DVD-RW Command-Line Syntax *(Continued)*

Argument	Description
imagesize={2MB,10MB,2GB,4GB}	.Specifies the image size used in write/read test
`erase={none, fast, all}`	.`none` - Do not erase media after test complete `fast` - Erase the last track added `all` - Erase entire disk
`eject`	Ejects the media.

CHAPTER **4**

Disk and Floppy Drives Test (`disktest`)

`disktest` verifies the functionality of hard drives and diskette drives using three subtests (see TABLE 4-1): Media, File System, and Asynchronous I/O.

Note – `disktest` does support x86 platforms on Solaris.

Most disk drives, such as SCSI disks, native or SCSI floppy disks, IPI, and so on, are supported. The type of drive being tested is displayed at the top of the Test Parameter Options dialog box.

Note – `disktest` is supported on x86 platforms that use the Solaris Operating System. For disks on x86 machines, the disk partitions could range from 0 to 15. `disktest` can be performed on any of these selected partitions.

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

`disktest` Test Requirements

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

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

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

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

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

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

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

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

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

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

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

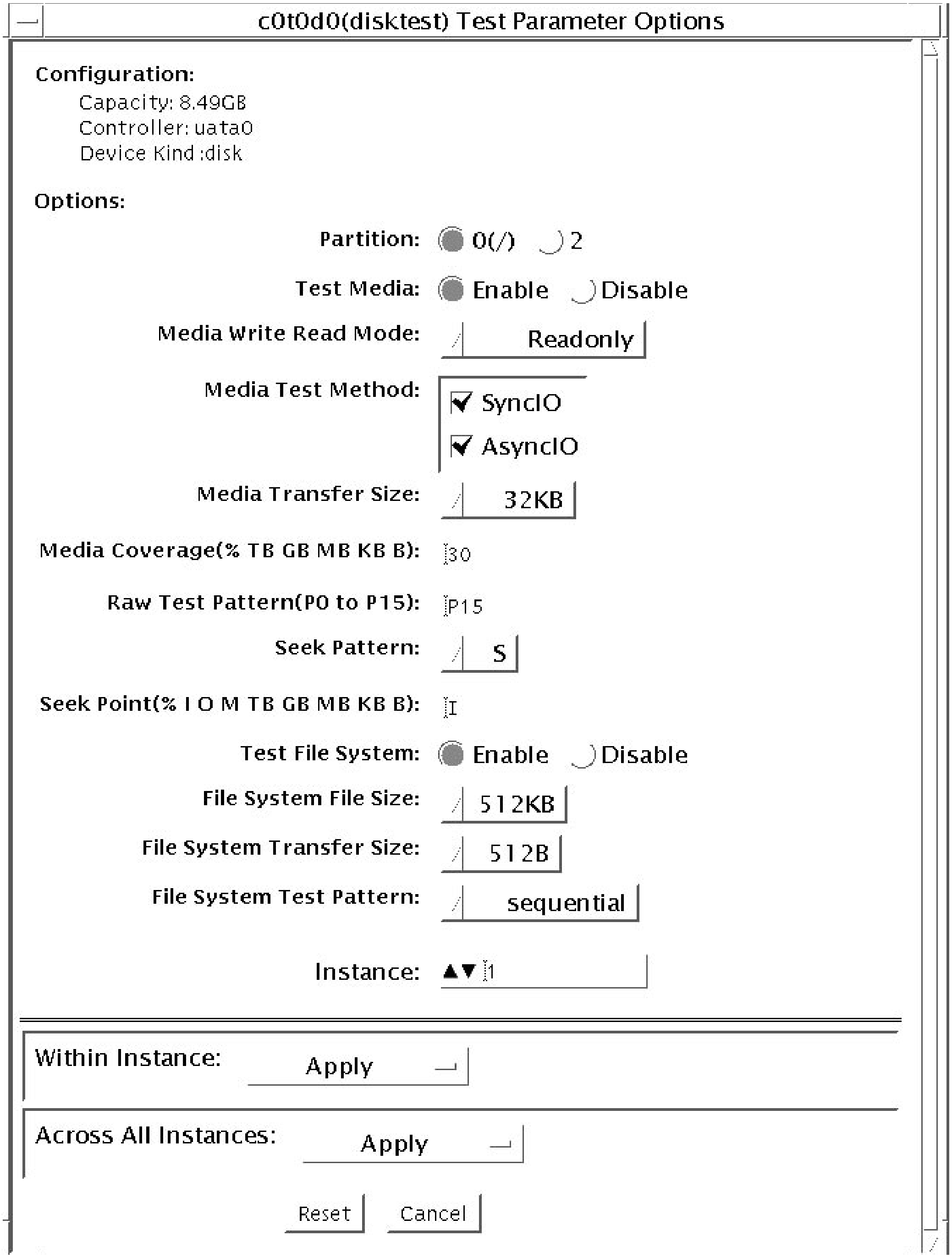
The following table describes the `disktest` subtests:

TABLE 4-1 `disktest` Subtests

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

`disktest` Test Options

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


c0t0d0(disktest) Test Parameter Options

Configuration:

Capacity: 8.49GB
Controller: uata0
Device Kind :disk

Options:

Partition: ● 0(/) ○ 2
Test Media: ● Enable ○ Disable
Media Write Read Mode: Readonly
Media Test Method: ☑ SyncIO ☑ AsyncIO
Media Transfer Size: 32KB
Media Coverage(% TB GB MB KB B): 30
Raw Test Pattern(P0 to P15): P15
Seek Pattern: S
Seek Point(% I O M TB GB MB KB B): I
Test File System: ● Enable ○ Disable
File System File Size: 512KB
File System Transfer Size: 512B
File System Test Pattern: sequential
Instance: ▲▼ 1

Within Instance: Apply
Across All Instances: Apply

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

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

TABLE 4-2 `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	Enable or Disable 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)	Enables users to test all or part of a partition (in percentage or in any of TB, GB, MB, KB, B units)
Raw Test Pattern (P0 to P16)	Enables user to specify the write, read pattern. • P0 – Low Frequency Pattern • P1 – Low Transition Density Pattern • P2 – High Transition Density Pattern • P3 – Compliant Jitter Pattern • P4 – Compliant Jitter: RPAT • P5 – Compliant Jitter: CRPAT • P6 – Compliant Jitter: JTPAT • P7 – Compliant Jitter: CJTPAT • P8 – Compliant Jitter: SPAT • P9 – Compliant Jitter: CSPAT • P10 – 8 Bit Cable Pattern • P11 – 16 Bit Cable Pattern • P12 – 8 Bit Xtalk Pattern • P13 – 16 Bit Xtalk Pattern • P14 – MFM Pattern • P15 – Generic Test Patterns • P16 – SATA Test Patterns
Seek Pattern	Enables specifying 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 4-2 `disktest` Configurations and Options *(Continued)*

`disktest` **Options**	**Description**
Seek Point (%, I, O, M, TB, GB, MB, KB, B)	Enables specifying the seek point offset for the I/O. You can specify the offset in percentage or any of TB, GB, MB, KB, B or and I, M, O; that is, 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 4-2 `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 4-3 `disktest` Supported Test Modes

Test Mode	Description
Connection	Only one instance of `disktest` (which monitors UNIX error messages) is allowed for each disk device. `disktest` displays messages and reports errors. The test also opens the hard disk, checks the disk configuration, reads a few blocks, and then closes the hard disk. No File System subtest is run. No Write option is available in Connection test mode.
Functional	More than one instance of `disktest` is allowed for one disk device. The File System subtest, Media subtests, and floppy test can be run in Functional test mode. In Functional mode, disktest performs 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>
["<(mount_point)>"], rawsub=E(nable)|D(isable), rawrw=
Readonly|CompareRead|WriteRead, rawiosize=<number>{...|KB|kb...}|random,
rawcover=<number>|<number>{TB|GB|MB|KB|B|tb|gb|mb|kb|b}
rawpattern=P(<0-16>)|0x<8 digit data pattern>, seekpattern=
{S|SR|LS|R|LB|LR|AB|AR}, seekpoint={i|m|o|<number>}, 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=***<data_pattern>*, **dev=***<device_name>*

TABLE 4-4 `disktest` Command-Line Syntax

Argument	Description
partition=*<0-7>* *["<(mount_point)>"***]**	Specifies the partition number as follows: • *n*—is the partition number (slice number), usually *0-7* for SPARC and *0-16* for x86 • *mount_point*—is the mount point for the mounted partition that you plan to test For example: **partition=6"(/export)"**
rawsub= *E(nable)|D(isable)*	Enables or disables the Media subtest. For example: **rawsub=** *Enable*
rawrw= *Readonly|CompareRead|WriteRead*	Specifies the Media subtest Read, Compare, and Write mode: • Read only • Read twice, Compare (works only with SyncIO method) • Write, Read, Compare, Restore For example: **rawrw=***ReadOnly*
rawiosize= *<number>{...|KB|kb...}|random*	Specifies the media size to transfer. The block size can be specified in kilobytes. For example: 2K,...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 units: TB, GB, MB, KB and B. For example: **rawcover=***40* OR **rawcover=***4GB*

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

Argument	Description
`rawpattern=`*P(<0-16>)* | *0x<8 digit data pattern>*	`rawpattern` could be specified as a pre-defined 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*
`seekpattern=` *{S | SR | LS | R | LB | LR | AB | AR}*	`seekpattern` could be specified to select the type of seek test to run on the disk drive. `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 exmaple: `seekpattern=`*S*

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

Argument	Description
`seekpoint=`*{i | m | o | <number>}*	Specify 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: a) `seekpoint=`*I*, start the I/O from block 1. b) `seekpoint=`*M*, start the I/O from middle offset of the partition.
`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.
`fspattern=`*<data_pattern>*	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: a) **`fspattern=`***0xa* a) **`fspattern=`***seq*
`fssize=` *<number>{K | KB | M | MB | k | kb | m | mb}*	Indicates the file system subtest size in Megabytes or Kilobytes: • `K|k|KB|kb` – kilobytes • `M|m|MB|mb` – megabytes `512KB|2MB| 8MB| 20MB|100MB|200MB`
`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`
`dev=`*device_name*	Specifies the name of the disk to be tested. For example: c0t3d0.

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

Note – 64-bit tests are located in the `/bin/64` directory, or the relative path in 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-Bit and 64-Bit Tests” on page 5.

CHAPTER **5**

Emulex HBA Test (`emlxtest`)

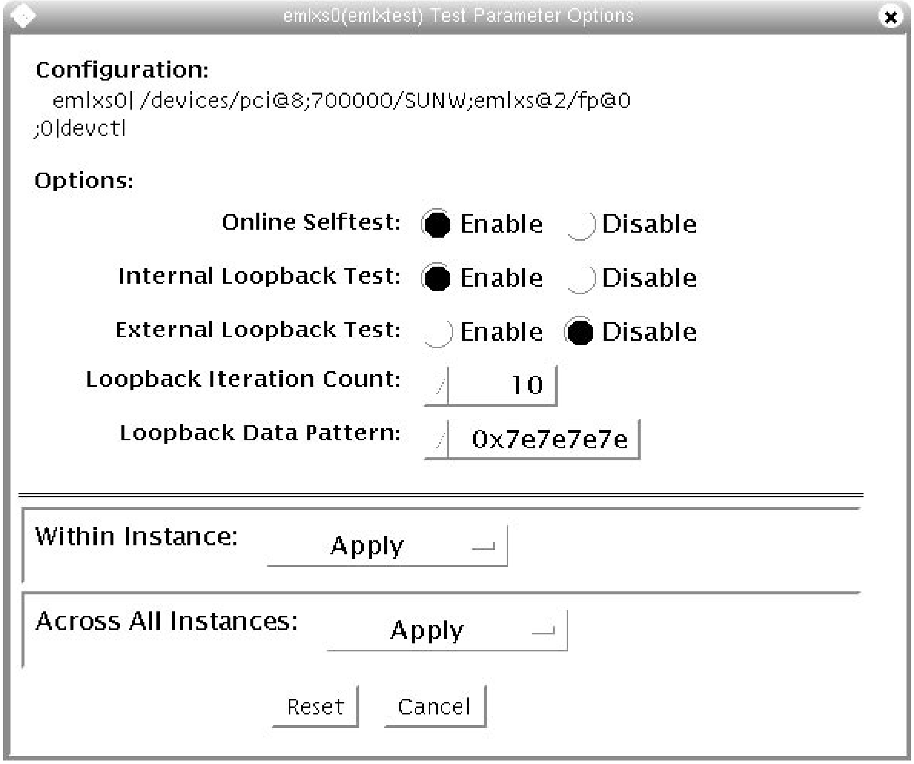
`emlxtest` verifies the proper functioning of the Emulex LP10000 Dual and Single Port 2 Gigabyte Fibre Channel PCI host bus adapters (HBAs). The supported HBAs are 133 MHz PCI-X 2 Gigabyte Fibre Channel based on the Emulex LP10000-S (Amber-2r) and LP10000DC-S (Crystal-2r) HBAs, which are the single and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

Note – `emlxtest` is supported on both Sun Ultra SPARC and x86 AMD64 architectures.

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.



Configuration:

emlxs0| /devices/pci@8;700000/SUNW;emlxs@2/fp@0
;0|devctl

Options:

Online Selftest: ● Enable ○ Disable
Internal Loopback Test: ● Enable ○ Disable
External Loopback Test: ○ Enable ● Disable
Loopback Iteration Count: 10
Loopback Data Pattern: 0x7e7e7e7e

Within Instance: Apply
Across All Instances: Apply

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

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

TABLE 5-1 `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 5-2 `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 5-3 `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.

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

CHAPTER **6**

Floating Point Unit Test (`fputest`)

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.

Note – `fputest` does support x86 platforms on Solaris.

Note – Three benchmarks of `fputest`—`cparanoia`, `kcdiv`, and `kcsqrt`—have been ported over for x86 systems. The x86 version of `fputest` only uses the Double Precision operations for rounding, chopping, sticky bit, and so on.

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:

- 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

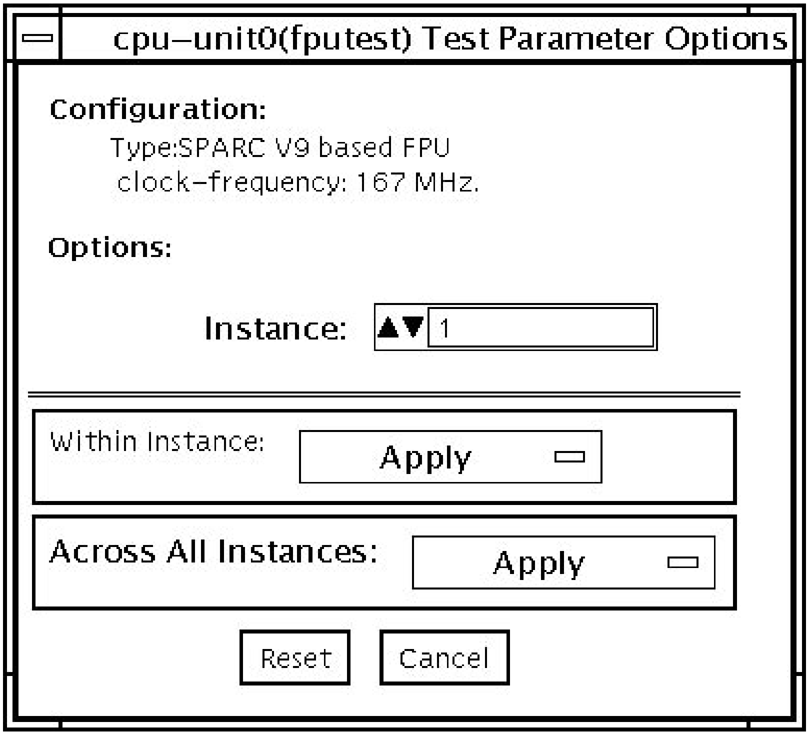
- Double-Precision Integer to Floating Point test
- Single-Precision Floating Point to Integer test
- Double-Precision Floating Point to Integer test
- Single-Precision Round Toward Zero test
- Double-Precision Round Toward Zero test
- Single to Double-Precision Format Conversion test
- Double to Single-Precision 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:

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

`fputest` Options

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



Type:SPARC V9 based FPU
clock–frequency: 167 MHz.

Instance: 1

Within Instance: Apply

Across All Instances: Apply

FIGURE 6-1 `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 6-1 `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 6-2 `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 6-2 `fputest` Command-Line Syntax

Argument	Description
`-n`	Enables connectivity 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
```

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-Bit and 64-Bit Tests" on page 3.

CHAPTER 7

Infiniband Host Channel Adapter Test (`ibhcatest`)

`ibhcatest` is comprised of multiple iRISC CPU cores, two 4x IB 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 IB fabric. Supported platforms include: two 1U and two 2U x86 AMD Opteron entry-level servers, Sun Fire V2xx, V4xx and E series high end servers.

Note – `ibhcatest` supports the same set of options for both SPARC and x86 platforms.

`ibhcatest` provides a mechanism to exercise and verify 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 goal is to isolate single fault to the identifiable component(s).

`ibhcatest` supports three execution test modes in SunVTS: Connection, Exclusive and Functional. In Connection mode, the test will provide a basic sanity check. This basic sanity test is done by querying for the Tavor firmware / 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 which 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. Also on the receiving side, data does not get verified by the CRC algorithm. Otherwise, all other components of Tavor that involve in transmitting and receiving data packets are being exercised.

Tavor based HCA is designed to use a single, 256 MB DDR memory for data storage at run time. This data storage is intended to be used and 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 atomic 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.

With these constraints in mind, the memory subtest is limited to read only operations to cover the entire DDR memory. There is no checking for data corruption and no mechanism for triggering single/double bit type of errors through writing to memory. The resulting benefit of read operations is a secondary effect that occurs in the generation of high volume PCI activities from the memory accesses. Thus the test becomes a good exerciser to bring out bus related problems.

`ibhcatest` Subtests

TABLE 7-1 `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. When transmitting a packet, if it is destined 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 should direct them to the destination port. This subtest uses interfaces from the Tavor driver to perform loopback testing. It is a push button type of test. 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 on/off the loopback test • `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); default: 55000; max: 1000000. • `loop=`*Number*: Number of loopback iterations for each pass; default: 200; max: 1000 • `warn=`*Enabled*|*Disabled*: When enabled, prints a warning message
DDR READ Test	This subtetst is comprised of 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 goes through and 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 the `random test` method, 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*: Turn off/off the DDR Memory test • `rdoffset=`*Offset*: Starting 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, default: 0x2000; max: 256 MB, in hexadecimal

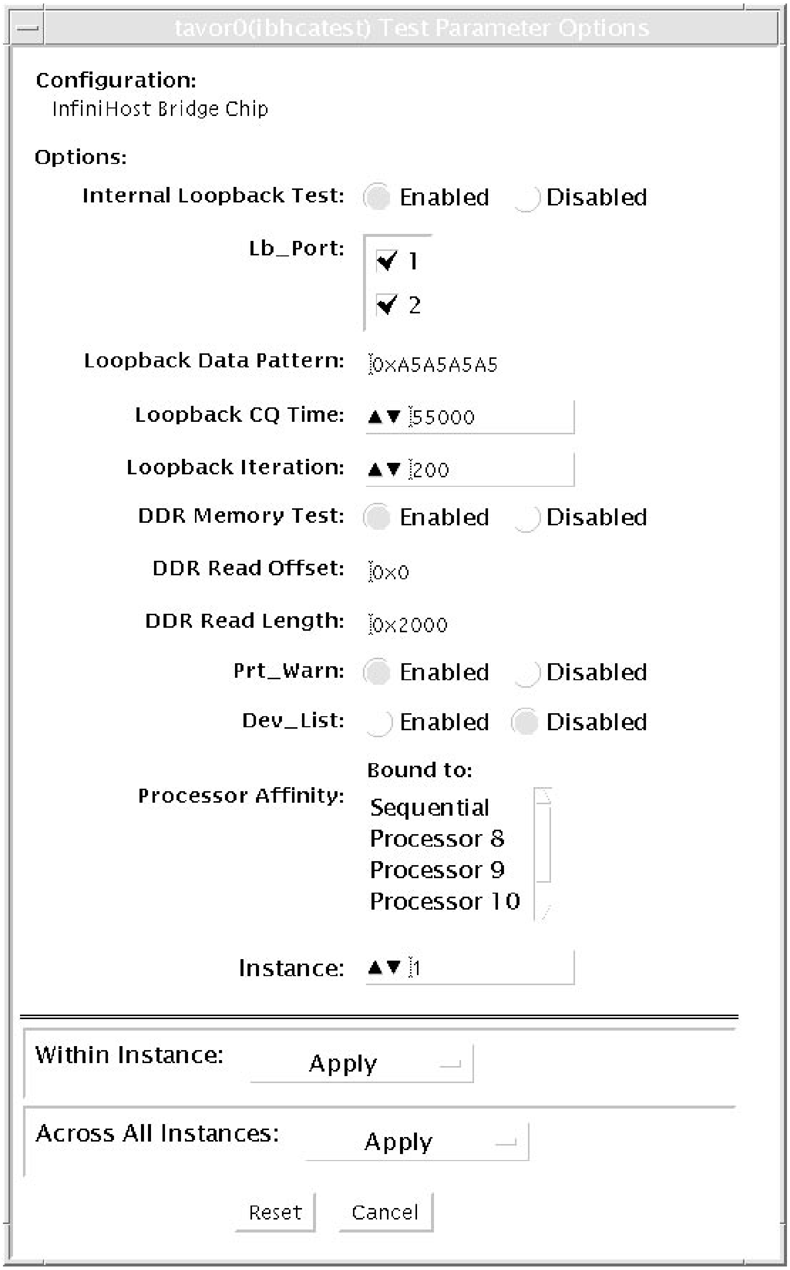
Caution – In SunVTS environment `ibhcatest` and `nettest` are mutually exclusive, `nettest` has higher priority if the IB port interface is plumbed up when SunVTS is invoked. These two tests can not be run at the same time at the command line, if both of these tests are invoked at the command line, `ibhcatest` 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 instance number of the interface.

`ibhcatest` Options

To reach the dialog box below, 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 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 may not include the device appropriate to this test. Refer to the SunVTS User's Guide.



tavor0(ibhcatest) Test Parameter Options

Configuration:
InfiniHost Bridge Chip

Options:
Internal Loopback Test: ● Enabled ○ Disabled
Lb_Port: ☑ 1 ☑ 2
Loopback Data Pattern: 0xA5A5A5A5
Loopback CQ Time: 55000
Loopback Iteration: 200
DDR Memory Test: ● Enabled ○ Disabled
DDR Read Offset: 0x0
DDR Read Length: 0x2000
Prt_Warn: ● Enabled ○ Disabled
Dev_List: ○ Enabled ● Disabled
Processor Affinity: Bound to: Sequential, Processor 8, Processor 9, Processor 10
Instance: 1

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

TABLE 7-2 `ibhcatest` Options

Option	Description
`lb=`*Enabled* | *Disabled*	Turn on/off the loopback test
`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), default: 55000; max: 1000000
`loop=`*Number*	Number of loopback iterations for each pass; default: 200; max: 1000
`ddr=`*Enabled* | *Disabled*	Turn on/off the DDR Memory test
`rdoffset=`*Offset*	Starting 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, default: 0x2000; max: 256 MB, in hexadecimal
`warn=`*Enabled* | *Disabled*	Print a warning message when enabled
`list`	Print device list, no testing when set

`ibhcatest` Test Modes

TABLE 7-3 `ibhcatest` Supported Test Modes

Test Mode	Description
Connection	Provides a basic sanity check by querying for the Tavor firmware / hardware revision and running internal loopback. The internal loopback test runs at least once depending on the amount of time each pass takes.
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 7-4 `ibhcatest` Command Line Syntax

Option	Description
`lb=`*Enabled* | *Disabled*	Turn on/off the loopback test
`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), default: 55000; max: 1000000.
`loop=`*Number*	Number of loopback iterations for each pass; default: 200; max: 1000
`ddr=`*Enabled* | *Disabled*	Turn on/off the DDR Memory test

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

Option	**Description**
`rdoffset=`*Offset*	Starting 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, default: 0x2000; max: 256 MB, in hexadecimal
`warn=`*Enabled* | *Disabled*	Print a warning message when enabled
`list`	Print device list, no testing when set

Note – 64-bit tests are located in the `/bin/64` directory, or the relative path in 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-Bit and 64-Bit Tests" on page 5.

CHAPTER **8**

Level 1 Data Cache Test (`l1dcachetest`)

`l1dcachetest` exercises the level1 Data cache in the CPU module of Sun systems. The test writes, reads, and verifies access of multiple virtual addresses. The virtual addresses are so chosen 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.

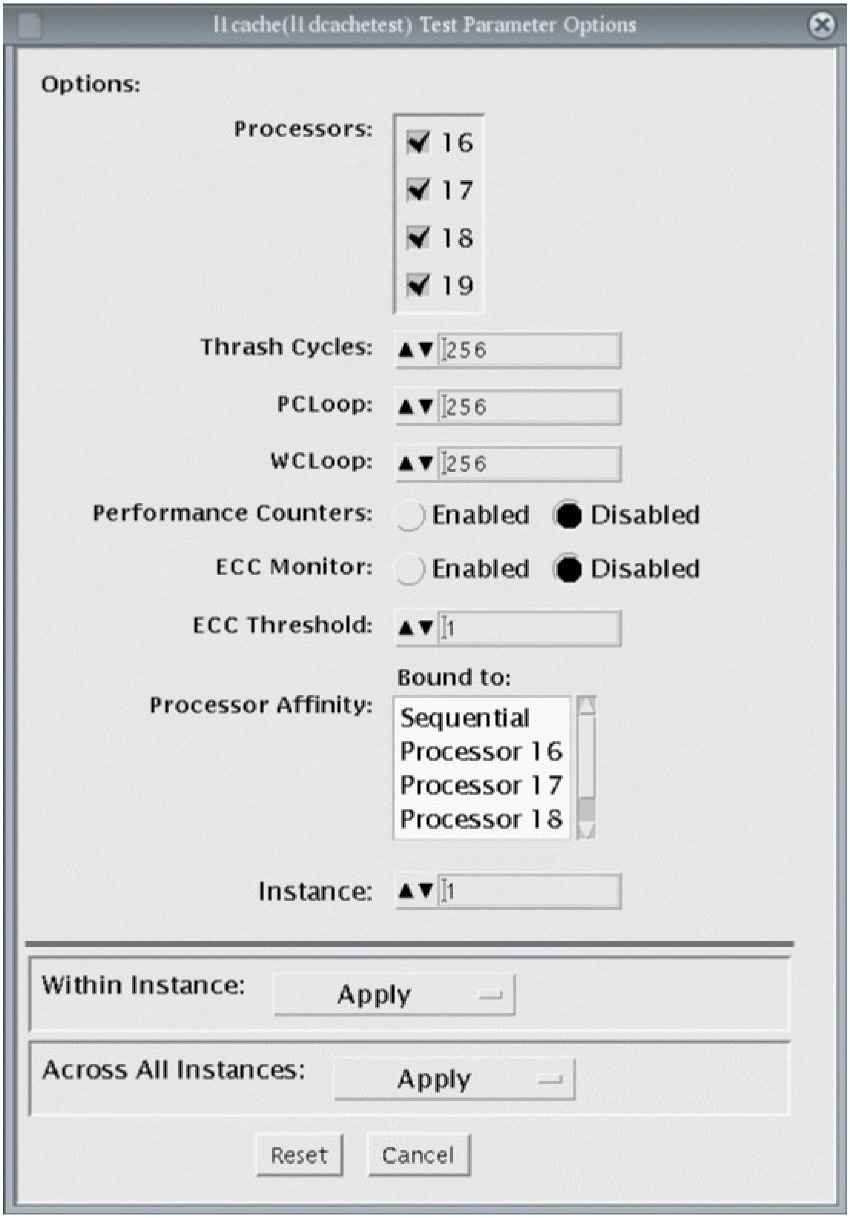
Note – `l1dcachetest` does support x86 platforms on Solaris.

`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, ensuring 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. It scales with the size of system. `l1dcachetest` is mult-threaded. Although, selection of CPU IDs, is kept as one of the options but if its not specified, it automatically retrieves the number of CPUs in the system and internally 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 dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.



l1cache(l1dcachetest) Test Parameter Options

Options:

Processors:
- ☑ 16
- ☑ 17
- ☑ 18
- ☑ 19

Thrash Cycles: 256

PCLoop: 256

WCLoop: 256

Performance Counters: ○ Enabled ● Disabled

ECC Monitor: ○ Enabled ● Disabled

ECC Threshold: 1

Processor Affinity: Bound to:
- Sequential
- Processor 16
- Processor 17
- Processor 18

Instance: 1

Within Instance: Apply

Across All Instances: Apply

FIGURE 8-1 `l1dcachetest` Test Parameter Options Dialog Box

TABLE 8-1 `l1dcachetest` Options

Option	Description
Processors	This option can be used to select the CPU IDs for which to run this test. The test will use all CPUs on the system by default. Hence this parameter is optional.
Thrash Cycles	Specifies the number of iteration 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 that come on the `/var/adm/messages` could be correctable errors, that is why the threshold value is provided to give facility to 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. By Default, it is OFF.

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

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

`l1dcachetest` Test Modes

TABLE 8-2 `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 8-3 `l1dcachetest` Command-Line Syntax

Argument	Description
`M=`*1+2+3...*	This option can be used to select the CPU IDs for which to run this test. The test will use all CPUs on the system by default. Hence 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 which is currently offline induces an appropriate error messages from the test. 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.

TABLE 8-3 `l1dcachetest` Command-Line Syntax

Argument	Description
`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 which could be caused during test. The default is OFF.
`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 that come on the `/var/adm/messages` could be correctable error, that is why threshold value is provided for the user to give facility to ignore the errors if they are below thrshold value. The default value is 1. If set to zero, the test will still report errors but will not stop.
`perf`= *Enabled | Disabled*	Enables or Diables the Performance counter measurements related to Data Cache and Prefetch Cache events. By Default, it is OFF.
`dev`=*l1cache*	Specifies the device. The default is l1cache.

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

Note – Command line syntax still supports old command line syntax, but this will not be supported in a fulure release of SunVTS.

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-Bit and 64-Bit Tests" on page 3.

CHAPTER 9

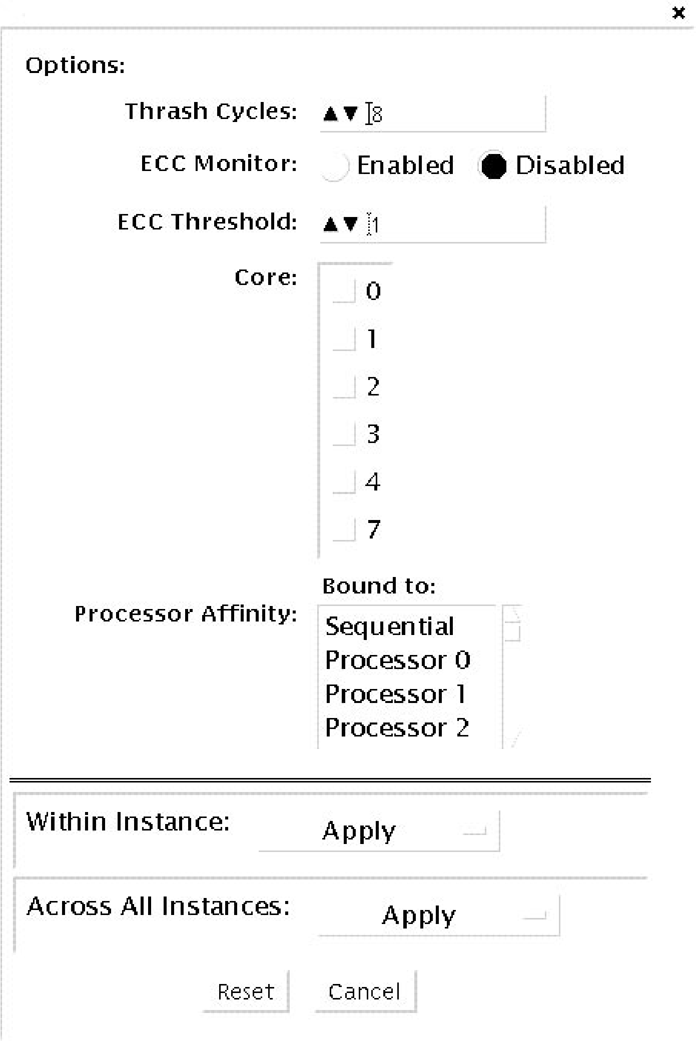
Level 2 Cache Test (`l2sramtest`)

`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/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 will automatically retrieve the number of CPUs in the system and internally create 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. The user does not have to input these values.

`l2sramtest` Options

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


Options:

Thrash Cycles: ▲▼ 8

ECC Monitor: ○ Enabled ● Disabled

ECC Threshold: ▲▼ 1

Core:

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 7

Processor Affinity: Bound to:

- Sequential
- Processor 0
- Processor 1
- Processor 2

Within Instance: Apply

Across All Instances: Apply

FIGURE 9-1 `l2sramtest` Test Parameter Options Dialog Box

TABLE 9-1 `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 9-2 `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 9-3 `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 **10**

Ethernet Loopback Test (`netlbtest`)

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.

Note – `netlbtest` does support x86 platforms on Solaris.

Note – For x86 platforms, `netlbtest` only supports network drivers that have been ported over to x86 Solaris platforms. `netlbtest` has been tested with Sun GigaSwift based cards on x86 systems.

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 will not be the same.

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

A new debugging capability has been added in `netlbtest`. After one packet is not received, four more packets are transmitted. If all of them 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 will pass.

`netlbtest` Test Requirements

You must have the Ethernet card and the device driver installed, a loopback connector in place, and Intervention mode enabled before running `netlbtest`. `netlbtest` cannot run and will 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
```

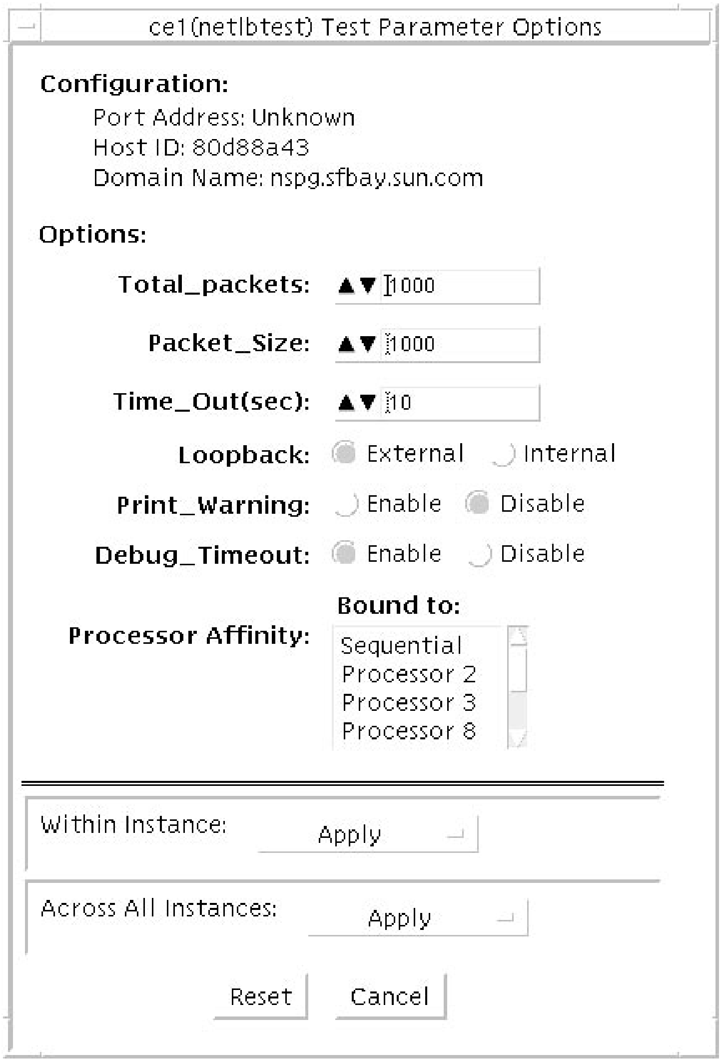
To run `netlbtest`, a loopback connector must be connected to the Ethernet interface. A loopback connector provides the network interface driver the necessary link for testing, while maintaining isolation from a live network. The loopback connector is required for both internal and external tests of the Ethernet device.

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



ce1(netlbtest) Test Parameter Options

Configuration:

Port Address: Unknown
Host ID: 80d88a43
Domain Name: nspg.sfbay.sun.com

Options:

Total_packets: 1000
Packet_Size: 1000
Time_Out(sec): 10
Loopback: ● External ○ Internal
Print_Warning: ○ Enable ● Disable
Debug_Timeout: ● Enable ○ Disable
Processor Affinity: Bound to: Sequential, Processor 2, Processor 3, Processor 8

Within Instance: Apply
Across All Instances: Apply

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

Refer to TABLE 10-1 for test parameter descriptions.

TABLE 10-1 `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.
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	Enable or disable the debugging feature of `netlbtest`. The default setting is Disable.

`netlbtest` Test Modes

TABLE 10-2 `netlbtest` Supported Test Modes

Test Mode	**Description**
Functional (Offline)	Runs the full set of subtests. It is assumed that the host is not connected to the network through the intended test device(s).

Since `netlbtest` requires a loopback connector, 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 10-3 `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. The maximum number of packets is 100,000,000.
`pksz=`*pkt_size*	[60... 1514], packet size in 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.

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-Bit and 64-Bit Tests" on page 4.

CHAPTER **11**

RAM Test (`ramtest`)

`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.

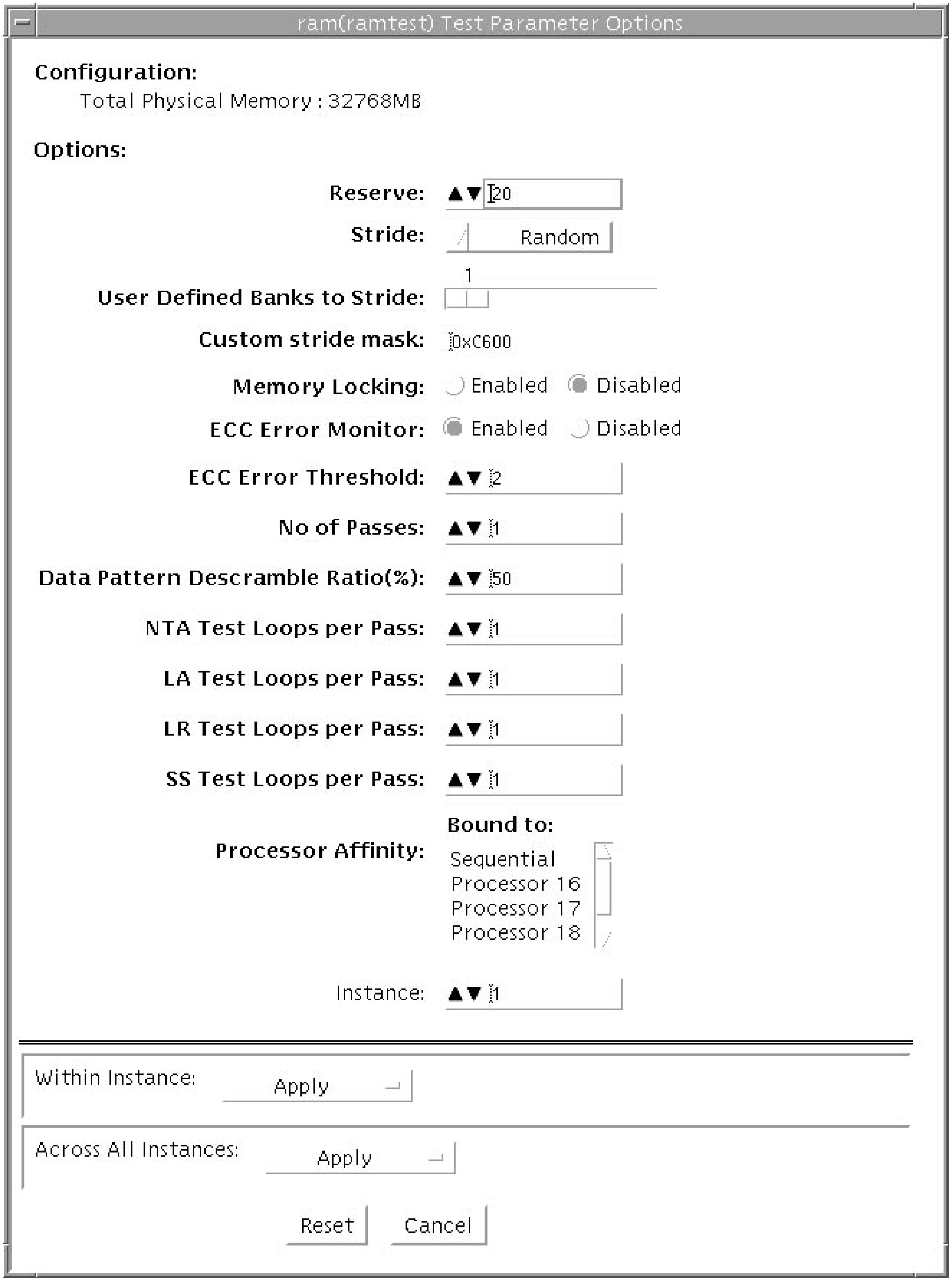
This test is being added only for the Exclusive mode testing because of the high stress it puts on the memory and the system interconnect. `ramtest` assumes that no other application is running at the same time.

Note – `disktest` is supported on x86 platforms that use the Solaris Operating System.

Caution – This is an Exclusive mode test. No other application should be running during this test.

`ramtest` Options

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



ram(ramtest) Test Parameter Options

Configuration:

Total Physical Memory : 32768MB

Options:

Reserve: 20

Stride: Random

User Defined Banks to Stride: 1

Custom stride mask: 0xC600

Memory Locking: ○ Enabled ● Disabled

ECC Error Monitor: ● Enabled ○ Disabled

ECC Error Threshold: 2

No of Passes: 1

Data Pattern Descramble Ratio(%): 50

NTA Test Loops per Pass: 1

LA Test Loops per Pass: 1

LR Test Loops per Pass: 1

SS Test Loops per Pass: 1

Processor Affinity: Bound to:
Sequential
Processor 16
Processor 17
Processor 18

Instance: 1

Within Instance: Apply

Across All Instances: Apply

FIGURE 11-1 `ramtest` Test Parameter Options Dialog Box

The following table details the `ramtest` options:

TABLE 11-1 `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, it is recommended to minimize swapping by tuning the reserve option. If for some reason 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. 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	Use this option to set the number of banks that the test should stride. One recommended choice is the interleave on the suspect bank, during FA. The value is currently limited to between 1 and 16. (This also means row striding is not possible while using this option).

TABLE 11-1 `ramtest` Options *(Continued)*

`ramtest` **Options**	**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 mixing row and column strides. Consult the Memory Controller section of the PRM for the CPU of the test system to discover 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.
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, this gives 4 MB virtual pages and avoids swapping. Running without locking on the other hand, adds more randomness to the addressing sequence.
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 on to the test output. The monitor can be turned off by setting this option to "Disabled". The ECC Monitor option is not supported on x86 platforms and an appropriate warning is displayed and the test proceeds based on other options.
ECC Error Threshold	This is the number of ECC errors after which the test will stop (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 of threshold is 2. The ECC Threshold option is not supported on x86 platforms and an appropriate warning is displayed and the test proceeds based on other options.
Number of Passes	This option specifies the number of passes, in the same instance. Increasing passes is recommended in case "lock" is enabled, this will save time spent on locking the memory every time a new process/instance is spawned by the VTS kernel. Note that this pass has no relation with the system passes in the VTS infrastructure, it will appear that `ramtest` is taking longer to complete system passes.

TABLE 11-1 `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 between cells in adjacent columns, or rows that are targeted. Note that 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. The SS March option is not supported on x86 platforms and an appropriate warning is displayed and the test proceeds based on other options.

`ramtest` Test Modes

TABLE 11-2 `ramtest` Supported Test Modes

Test Mode	Description
Exclusive	Generates enormous amount of memory traffic.

`ramtest` Command-Line Syntax

`/opt/SUNWvts/bin/sparcv9/ramtest` *standard_arguments* **`[ -o`**

`[ 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 11-3 `ramtest` Command-Line Syntax

Argument	Description
`reserve`	This is used to specify 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 US IIIi platforms, default value is tuned to 25. It should be noted that on low memory systems, the reserve value should be kept 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. 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 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 right now. (This also means row striding is not possible while using this option).
`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 mixing row and column strides. Consult the Memory Controller section of the PRM for the CPU of the test system to discover 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.

TABLE 11-3 `ramtest` Command-Line Syntax

Argument	Description

`lock`	By default memory locking is "Disabled". To turn it on set 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 on the other hand, adds more randomness to the addressing sequence.
	It should be noted that 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". The ECC Monitor option is not supported on x86 platforms and an appropriate warning is displayed and the test proceeds based on other options.
`threshold`	This is the number of ECC errors after which the test will stop (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. The ECC Threshold option is not supported on x86 platforms and an appropriate warning is displayed and the test proceeds based on other options.

TABLE 11-3 `ramtest` Command-Line Syntax

Argument	Description
`pass`	This option specifies number of passes, in the same instance. Increasing pass is recommended in case "lock" is enabled, this will save time spent on locking the memory every time a new process/instance is spawned by the VTS kernel. Note that this pass has no relation with the system passes in the VTS 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 11-3 `ramtest` Command-Line Syntax

Argument	Description
`dratio`	Descrambles ratio can be used to tune the algorithm used to generate data patterns in `ramtest`. Descramble ratio of 100 means that all the data patterns generated will be descrambled. Where as if descramble ratio is 0, the test will generate the data patterns tuned towards 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. The SS March option is not supported on x86 platforms and an appropriate warning is displayed and the test proceeds based on other options.
`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. Consult the Memory Controller section of the PRM for the CPU of the test system to discover 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.

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

Note – ECC errors returned by `ramtest` are actually detected by the operating system and are logged in the `/var/adm/messages` file. Please review this file for more detailed information regarding errors.

Note – 64-bit tests are located in the `/bin/64` directory, or the relative path in which you installed SunVTS. If a test is not present in this directory, then it might be available as a 32-bit test only.

CHAPTER **12**

System Test (`systest`)

The `systest` checks the overall functionality of a Sun system by exercising the CPU, I/O, and memory channels.

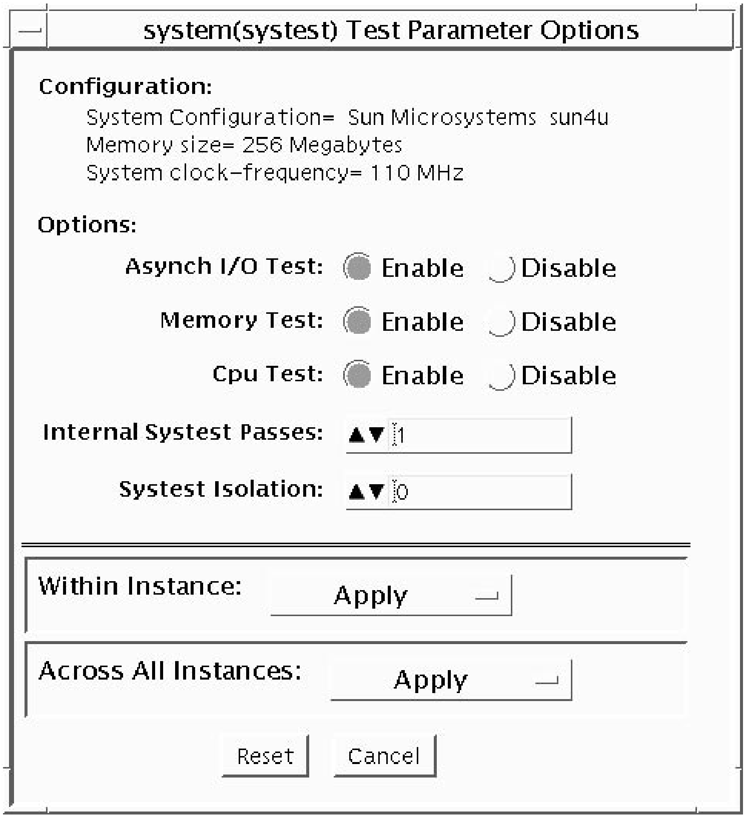
Note – `systest` does support x86 platforms on the Solaris OS.

Note – `systest` has been enhanced to enable the CPU test subtest on x86 systems. `systest` now supports the same set of options for both SPARC and x86.

The test ensures the concurrency of the different channels by the use of Solaris threads. The test aims at stimulating failures that might be caused due to the interaction of the various different hardware modules in the system. It is very stressful on the CPU, and stresses the parallel computational capability of a multiprocessor system.

`systest` Options

To reach the 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.


system(systest) Test Parameter Options

Configuration:

System Configuration= Sun Microsystems sun4u
Memory size= 256 Megabytes
System clock-frequency= 110 MHz

Options:

Asynch I/O Test: ● Enable ○ Disable
Memory Test: ● Enable ○ Disable
Cpu Test: ● Enable ○ Disable
Internal Systest Passes: ▲▼ 1
Systest Isolation: ▲▼ 0

Within Instance: Apply
Across All Instances: Apply

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

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

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

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

TABLE 12-1 `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 lapack passes. A set of boards and CPUs will be declared GOOD after `syspass` number of passes. The default is 1.
System Isolation	Defines the type of Isolation that `systest` needs to perform if a residual error is found in the initial evaluation phase. 0 = No Isolation (default) 1 = Board(s) Isolation only 2 = Board(s) and CPUs Isolation

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

`systest` Test Modes

TABLE 12-2 `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 12-3 `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. The CPU Test is not supported on x86 platforms.
`dev=`*system*	Specifies the pseudo device name.
`syspass=`*1,2000*	Defines the number of internal lapack passes. A set of boards and CPUs will be declared "GOOD" after “`syspass`” number of passes. The default is 1.
`sysiso=`*0*|*1*|*2*	Defines the type of Isolation that `systest` needs to perform if a residual error is found in the initial evaluation phase. 0 = No Isolation 1 = Board(s) Isolation only 2 = Board(s) and CPUs Isolation

---

Note – 64-bit tests are located in the `sparcv9` subdirectory: `/opt/SUNWvts/bin/sparcv9/`*testname*, or the relative path to which you installed SunVTS. If a test is not present in this directory, then it might be available as a 32-bit test only. For more information, see “32-Bit and 64-Bit Tests” on page 3.

Recommended Option Selection

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

Command-Line Examples

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

Example 1:

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

The above example invokes the following:

- `systest` with default parameter values.
- I/O, MEM, and CPU subtests. The CPU test is not supported on x86 platforms.
- One internal pass of lapack and no Isolation.

Example 2:

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

The above example invokes the following:

- `systest` without the I/O subtest
- MEM and CPU subtests. The CPU test is not supported on x86 platforms.
- One internal pass of lapack and no Isolation.

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

Example 3:

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

The above example invokes the following:

- I/O, MEM, and CPU subtests. The CPU test is not supported on x86 platforms.
- Declares a set of boards free from residual errors after 15 internal passes of the lapack algorithm.
- If an error is found, `systest` will perform boards isolation.

Example 4:

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

The above example invokes the following:

- I/O, MEM, and CPU subtests. The CPU Test is not supported on x86 platforms.
- Declares a set of boards AND CPUs free from residual errors after 10 internal passes of the lapack algorithm
- If an error is found, `systest` will perform boards AND CPUs isolation.

CHAPTER **13**

Tape Drive Test (`tapetest`)

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.

`tapetest` Test Requirements

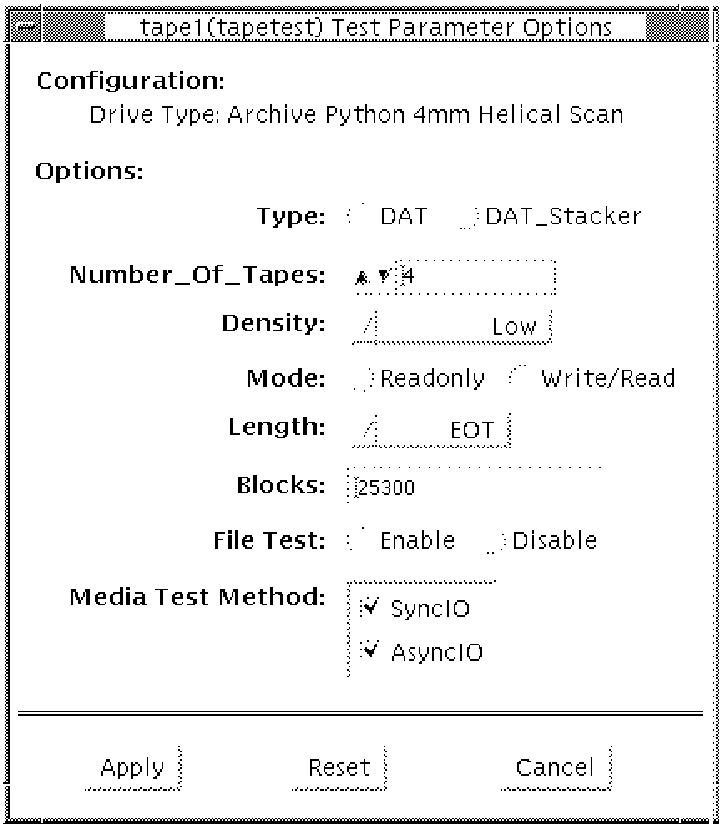
If you have a tape drive in your system, load a blank writable tape (scratch tape) before you start SunVTS. If you fail to do this, the `tapetest` option may display `drive type:unknown` on the option menu for the `tapetest`.

`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 may 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 Options dialog box for a device is shown in FIGURE 13-1.

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.



tape1(tapetest) Test Parameter Options

Configuration:
Drive Type: Archive Python 4mm Helical Scan

Options:
Type: DAT DAT_Stacker
Number_Of_Tapes: 4
Density: Low
Mode: Readonly Write/Read
Length: EOT
Blocks: 25300
File Test: Enable Disable
Media Test Method: ☑ SyncIO ☑ AsyncIO

FIGURE 13-1 `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 Options dialog box for the 1/4-inch, 1/2-inch, DLT, and 8-mm tape drives differ slightly from FIGURE 13-1.

TABLE 13-1 `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.

TABLE 13-1 `tapetest` Options

`tapetest` **Options**	**Description**
# of Blocks	If you select Specified under the Length option, you must type the number of blocks you want to test.
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. Note1: This option is only available in command line interface mode. Note2: 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 13-2 `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 (Offline)	`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 13-3 `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 13-3 `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

---

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-Bit and 64-Bit Tests" on page 3.

CHAPTER 14

USB Device Test (`usbtest`)

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` is a newly consolidated test which is used to test the USB ports using various USB devices such as the USB key board, the USB printer or the USB audio devices. `usbtest` supports the same set of options for both SPARC and x86.

For keyboard, the test verifies whether the keyboard(s) attached to the USB bus are USB compliant. The test will flash the LEDs of a compliant keyboard. The usbtest for printer 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

Subtests for the USB audio device:

- Tone subtest

  The Tone subtest is only performed in the Connectivity test mode. This subtest is user-interactive in that the user must listen when the test is performed. This subtest generates two seconds of sound which is output to the speakers.

- Record/Play subtest

This subtest performs a simple check that records and plays one second of random data at 8kHz sampling. It simply reads random data from the USB microphone port and plays back to the USB speakers.

- Audio subtest

  This subtest plays a 30 second music file which is output to the speakers. This is a partially user-interactive test. If there are no system call errors, the user must decide, by listening, if the test passed or failed. Things to listen for are distortion or lack of music.

Subtests for the USB printer device:

- getdevid subtest

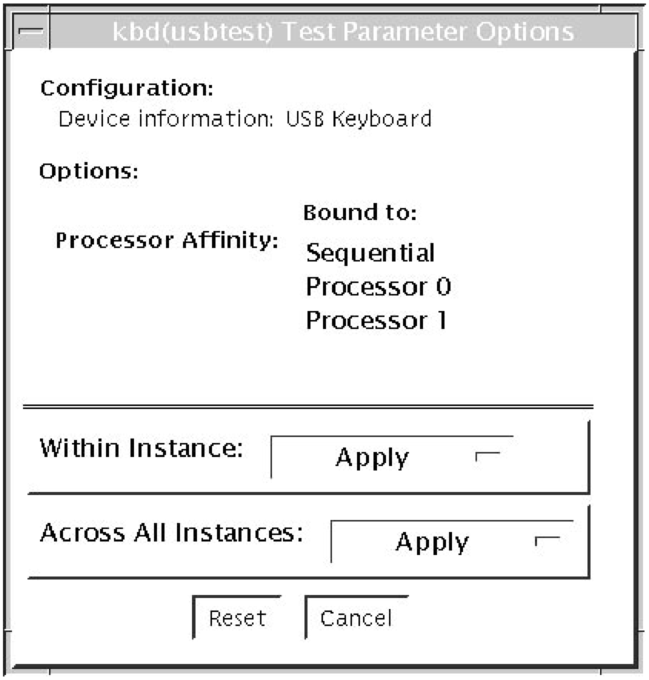
  The getdevid subtest retrieves the IEEE 1284 ID string of the printer.

- Printer subtest

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



kbd(usbtest) Test Parameter Options

Configuration:

Device information: USB Keyboard

Options:

Processor Affinity: Bound to: Sequential, Processor 0, Processor 1

Within Instance: Apply

Across All Instances: Apply

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

TABLE 14-1 `usbtest` Opions

Option	**Description**
Options available for USB audio devices:	
Volume	Specifies the volume of 0 to 255; the default is 80
Audiotest	Can be set to "enable" or "disable" to perform the audio music test; the default is "enable"
Option available for 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 14-2 `usbtest` Supported Test Modes

Test Mode	Description
Connection	For a USB audio device, the test outputs two seconds of sound to the speakers For 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. The user can enable Audio Music test For USB keyboard device, the test runs the full test For 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 14-3 `usbtest` Command Line Syntax

Argument	Description
`dev=`*dev-name*	The target device name. The test runs on the specified device name. The default device name is `/dev/kbd`.
Options for the USB printer	
`getdevid=`*Enable*|*Disable*	The option is available only for USB printer device. Enables or disables getdevid subtest. The subtest is enabled by default.
`printer=`*Enable*|*Disable*	Enables or disables the printer test. The subtest is disabled by default. The subtest requires a printer to be attached to the USB port.

TABLE 14-3 `usbtest` Command Line Syntax *(Continued)*

Argument	Description
`data=`*ascii*	Choose whether ASCII text or Postscript data is sent to the printer.
`delay=`*0-86400*	Allows you to choose a delay between passes of the printer test.
Options for the USB audio device	
`M=`*Enable | Disable*	Enables or disables the music play test. The default is enable.
`O=`*speaker | headphone | line-out*	Selects the output device the test needs to use.
`V=`*0-255*	Music output volume; the default is 80.
`MF=`*musicfilename*	Selects the music file; the default is `music.au`.