



# SunVTS™ 6.3 Patch Set 2 Documentation Supplement for SPARC® Platforms

---

Sun Microsystems, Inc.  
www.sun.com

Part No. 820-3164-10  
September 2007, Revision A

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

Copyright 2007 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, Java, 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 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, Californie 95054, États-Unis. Tous droits réservés.

Sun Microsystems, Inc. possède les droits de propriété intellectuelle relatifs à la technologie décrite dans ce document. En particulier, et sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plusieurs des brevets américains listés sur le site <http://www.sun.com/patents>, un ou les plusieurs brevets supplémentaires ainsi que les demandes de brevet en attente aux États-Unis et dans d'autres pays.

Ce document et le produit auquel il se rapporte sont protégés par un copyright et distribués sous licences, celles-ci 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.

Tout logiciel tiers, sa technologie relative aux polices de caractères, comprise, est protégé par un copyright et licencié par des fournisseurs de Sun.

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

Sun, Sun Microsystems, le logo Sun, Java, AnswerBook2, docs.sun.com, et Solaris sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux États-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 États-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface utilisateur 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 dans la recherche et le développement du concept des interfaces utilisateur visuelles ou graphiques pour l'industrie informatique. Sun détient une licence non exclusive de Xerox sur l'interface utilisateur graphique Xerox, cette licence couvrant également les licenciés de Sun implémentant les interfaces utilisateur graphiques OPEN LOOK et se conforment en outre aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DÉCLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES DANS LA LIMITE DE LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE À LA QUALITÉ MARCHANDE, À L'APTITUDE À UNE UTILISATION PARTICULIÈRE OU À L'ABSENCE DE CONTREFAÇON.



# Contents

---

**Preface** v

- 1. SunVTS Overview** 1
  - SunVTS Version Information 2
  - SunPCi-3 Card Support 2
  - Test Requirements 2
  - Collection of SunVTS Tests 3
    - 32- and 64-Bit Tests 3
  - SunVTS User Interfaces 4
    - Running a Test From a User Interface 4
    - Running a Test From the Command Line 6
  - Frame Buffer Tests 8
    - Testing Multiple Frame Buffers 9
    - Remote Testing of Frame Buffers 9
    - Locking Frame Buffers 10
- 2. CPU Test (cputest)** 11
- 3. Integer Unit Test (iutest)** 13

<b>4. Ethernet Loopback Test (netlbtest)</b>	<b>15</b>
netlbtest Description	15
netlbtest Test Requirements	16
netlbtest Options	17
netlbtest Test Modes	19
netlbtest Command-Line Syntax	20

# 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.

This document is a supplement to the *SunVTS 6.3 Test Reference Manual for SPARC Platforms* (820-0081) and the *SunVTS 6.3 User's Guide* (820-0080). Refer to these documents for details on the full collection of SunVTS 6.3 tests and software features that are not new or enhanced in this release.

---

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

---

---

## Before You Read This Book

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

- *SunVTS 6.3 User's Guide*
- *SunVTS 6.3 Test Reference Manual for SPARC Platforms*
- *SunVTS Quick Reference Card*

---

# How This Document Is Organized

[Chapter 1](#) provides a brief overview of SunVTS and describes new or enhanced features for this release.

The remaining chapters are actual test reference chapters of SunVTS tests that incurred changes for this release. These chapters are extracted from the SunVTS 6.3 Test Reference Manual.

---

# 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	<i>machine-name%</i>
C shell superuser	<i>machine-name#</i>
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 <code>.login</code> file. Use <code>ls -a</code> to list all files. % You have mail.
<b>AaBbCc123</b>	What you type, when contrasted with on-screen computer output	% <b>su</b> password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this. To delete a file, type <code>rm filename</code> .

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

---

## Related Documentation

This document covers SunVTS version 6.3 tests. It serves as a reference companion to the SunVTS documents listed below.

Application	Title	Part Number
Installation and Navigation	<i>SunVTS 6.3 User's Guide</i>	820-0080
Test Specific Reference	<i>SunVTS 6.3 Test Reference Manual for SPARC Platforms</i>	820-0081
Quick Reference Card	<i>SunVTS Quick Reference Card</i>	820-0083

---

## 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 6.3 Patch Set 2 Documentation Supplement for SPARC Platforms*, part number 819-7446-10

## SunVTS Overview

---

This chapter contains the following topics:

- [“Test Requirements” on page 2](#)
- [“Collection of SunVTS Tests” on page 3](#)
- [“SunVTS User Interfaces” on page 4](#)
- [“Frame Buffer Tests” on page 8](#)

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

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

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

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

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

---

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

---

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

---

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

---

## SunVTS Version Information

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

## SunPCi-3 Card Support

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

---

## Test Requirements

SunVTS 6.3 was first introduced and designed to run in the Solaris 10 1/06 (Solaris 10 Update 1) OS and subsequent compatible releases. SunVTS 6.3 is not supported on OS releases prior to Solaris 10 3/05 (Solaris 10).

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

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

---

# Collection of SunVTS Tests

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

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

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

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

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

## 32- and 64-Bit Tests

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

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

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

```
# isainfo -v
```

---

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

---

---

# SunVTS User Interfaces

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

**TABLE 1-1** SunVTS System Interfaces

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

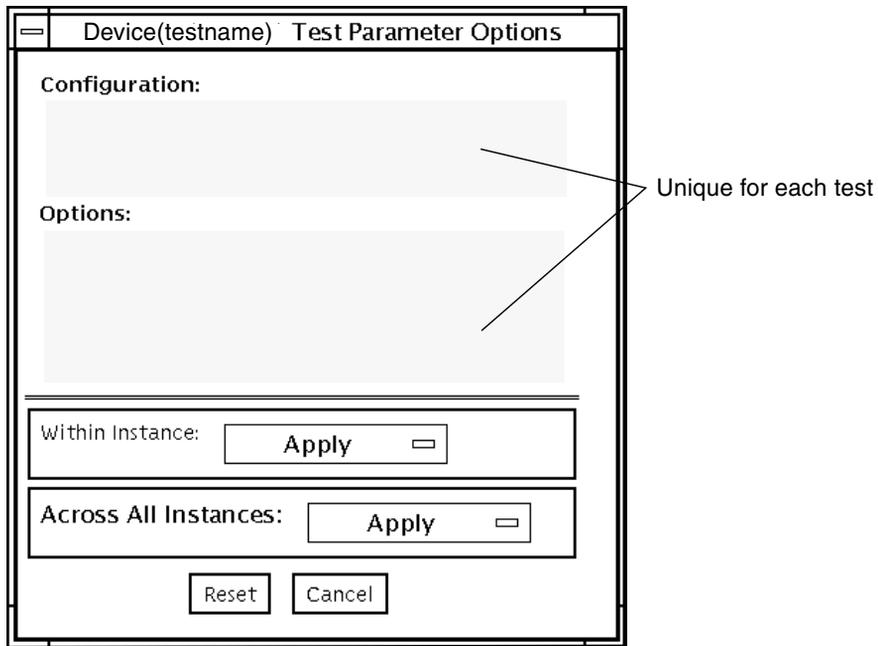
## Running a Test From a User Interface

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

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

### Test Parameter Options Dialog Box

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



**FIGURE 1-1** Test Parameter Options Dialog Box

**TABLE 1-2** Test Parameter Options Dialog Box Items

Menu Item	Description
Configuration	Information such as device type, capacity, revision, and serial numbers for the selected device. This information cannot be changed.
Options	A list of test options that are used to customize the testing of the selectable device, group, or all devices. The options are specific for each test and are covered in the test specific-chapters in this manual.
Within Instance	Provides the means to apply the settings: <ul style="list-style-type: none"> <li>• To this device only with Apply</li> <li>• To all devices within this group with Apply to Group</li> <li>• To all devices (of the <i>same device type for all controllers</i>) with Apply to All</li> </ul> <p>The option settings are only applied to one instance of the test.</p>

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

Menu Item	Description
Across All Instances	Provides the means to apply the settings globally: <ul style="list-style-type: none"><li>• To this device only with Apply</li><li>• To all devices within this group with Apply to Group</li><li>• To all devices (of the <i>same device type</i> for all controllers) with Apply to All</li></ul> The option settings are applied to all instances.
Reset	Returns the option values to their default settings and closes the Test Parameter Option dialog box.
Cancel	Ignores any changes made to option values and closes the Test Parameter Option dialog box.

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

## Running a Test From the Command Line

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

Unless specified, the test runs without the SunVTS kernel (`vt.sk`). 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 [-scruidtelnf] [-i number] [-w number] [-o test-specific-arguments]
```

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

## Standard Command-Line Arguments

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

**TABLE 1-3** Standard Command-Line Arguments

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

---

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

---

---

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

---

## Test-Specific Arguments

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

---

## Frame Buffer Tests

SunVTS includes a number of tests that exercise frame buffers:

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

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



---

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

---



---

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

---



---

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

---

## Testing Multiple Frame Buffers

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

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

## Remote Testing of Frame Buffers

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

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

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

# Locking Frame Buffers

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

## ▼ To Enable Frame Buffer Locking

- **Take one of the following actions:**

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

For example:

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

## ▼ To Disable Frame Buffer Locking

- **Take one of the following actions:**

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

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

## CPU Test (cputest)

---

Starting from 6.3ps1, the `cputest` registers only one instance for the whole system. The parameter `dev`, which used to take option of the form `dev=cpu-unitx`, is enhanced to support testing multiple CPUs present on the system. The option `dev` now takes the following syntax:

```
dev=all|All|cpu-unit0+cpu-unit1+.....+cpu-unitN
```

The `dev` option also supports either “all” or “All” as a valid value in which case all the CPUs in the system can be tested.



## Integer Unit Test (`iutest`)

---

Starting from 6.3ps1, the `iutest` registers only one instance for the whole system. The parameter `dev`, which used to take option of the form `dev=cpu-unitx`, is enhanced to support testing multiple CPUs present on the system. The option `dev` now takes the following syntax:

```
dev=all|All|cpu-unit0+cpu-unit1+.....+cpu-unitN
```

The `dev` option also supports either “all” or “All” as a valid value in which case all the CPUs in the system can be tested.



## Ethernet Loopback Test (netlbtest)

- 
- “netlbtest Description” on page 15
  - “netlbtest Test Requirements” on page 16
  - “netlbtest Options” on page 17
  - “netlbtest Test Modes” on page 19
  - “netlbtest Command-Line Syntax” on page 20

---

**Note** – The netlbtest external loopback test is not supported in 1000Mbps mode for the Sun PCI-Express Dual Gigabit Ethernet MMF/UTP Adapter and the e1000g driver.

---

---

### netlbtest Description

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

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

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

---

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

---

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

---

## netlbttest Test Requirements

You must have the Ethernet card and the device driver installed, a loopback connector in place (external loopback only), and Intervention mode enabled before running `netlbttest`. `netlbttest` cannot run and does not appear in the GUI if the network interface is connected to a live network. `netlbttest` 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 `netlbttest`. Enter the following commands to bring the interface down:

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

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

---

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

---

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

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

---

## netlbtest Options

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

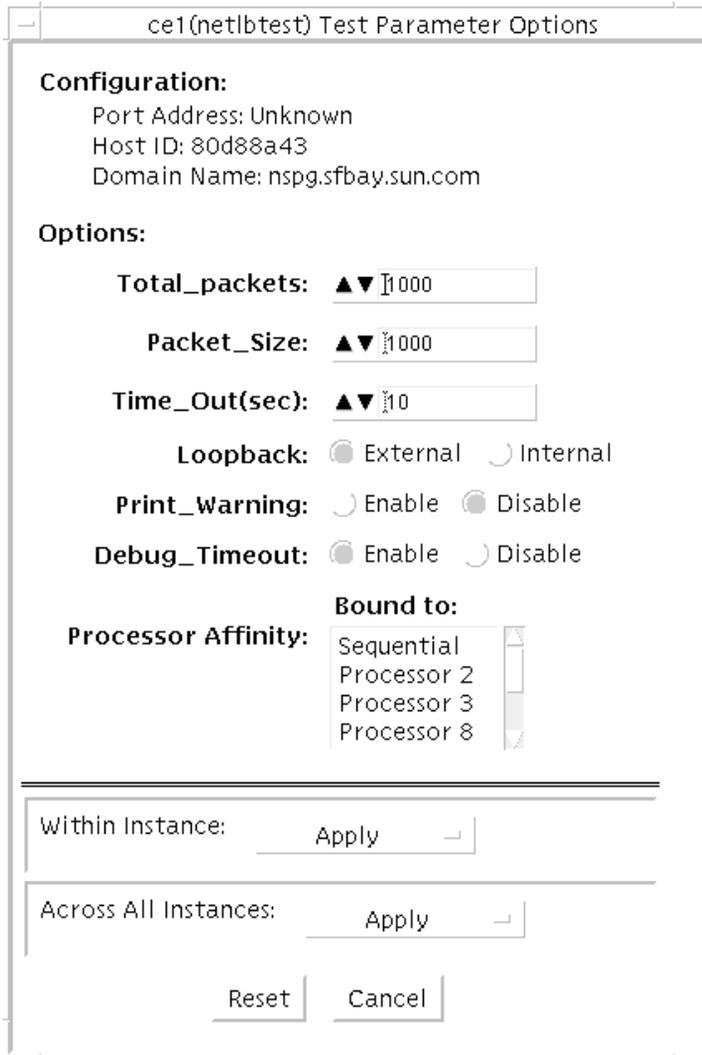


FIGURE 4-1 netlbtst Test Parameter Options Dialog Box

Refer to [TABLE 4-1](#) for test parameter descriptions.

**TABLE 4-1** 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. $60 \leq \text{packet size} \leq 1514$ . The default packet size is 1000 bytes.
Time_Out(sec)	Determines the amount of time (in seconds) that netlbtest can wait to receive packets. If no packets are received within this time frame, netlbtest reports an error message. The range for timeout is from 1 to 1,000 seconds.
Loopback	Determines the external and internal loopback mode. The default setting is internal loopback mode.
Print_Warning	Enables or disables the printing of warning messages. The default setting is Disable.
Processor Affinity	Binds the test to a specific processor. If no processor is specified, the test migrates between processors. This option is only available on multiprocessor systems.
Debug_Timeout	Enables or disables the debugging feature of netlbtest. The default setting is Disable. The range for timeout is from 1 to 1,000 seconds.

---

## netlbtest Test Modes

**TABLE 4-2** netlbtest Supported Test Modes

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

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

---

# netlbttest Command-Line Syntax

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

**TABLE 4-3** netlbttest Command-Line Syntax

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