Contents

Preface xvii

1. Introduction 1
   Test Requirements 2
   Collection of SunVTS Tests 2
       32-Bit and 64-Bit Tests 3
   SunVTS User Interfaces 3
       Running a Test From a User Interface 4
       Running a Test From the Command Line 6
   Testing Frame Buffers 8
       Testing Multiple Frame Buffers 9
       Remote Testing of Frame Buffers 10

2. Advanced Frame Buffer Test (afbtest) 11
   afbtest Test Requirements 11
   afbtest Options 12
   afbtest Test Modes 19
   afbtest Command-Line Syntax 20

3. Alarm Card Test (alarmtest) 23
   alarmtest Subtests 23
14. **DVD Test** *(dvdtest)* 95
   - dvdtest Test Requirements 95
   - dvdtest Options 95
   - dvdtest Test Modes 97
   - dvdtest Command-Line Syntax 97

15. **ECP 1284 Parallel Port Printer Test** *(ecpptest)* 99
   - ecpptest Hardware and Software Requirements 99
   - ecpptest Subtests 100
   - ecpptest Options 100
   - ecpptest Test Modes 103
   - ecpptest Command-Line Syntax 103

16. **Sun StorEdge A5x00 Test** *(enatest)* 105
   - enatest Options 109
   - enatest Fault Isolation Capability 112
   - enatest Test Modes 113
   - enatest Command-Line Syntax 113

17. **Sun StorEdge 1000 Enclosure Test** *(enctest)* 115
   - enctest Options 116
   - enctest Test Modes 118
   - enctest Command-Line Syntax 118

18. **Environmental Test** *(envtest)* 119
   - envtest Options 119
   - envtest Test Modes 122
   - envtest Command-Line Syntax 122

19. **Environmental Test** *(env2test)* 123
   - env2test Options 123
20. **Environmental Test (env3test)** 127
   - env3test Options 127
   - env3test Test Modes 129
   - env3test Command-Line Syntax 129

21. **Environmental Test (env4test)** 131
   - env4test Options 132
   - env4test Test Modes 135
   - env4test Command-Line Syntax 135

22. **Environmental Test (env5test)** 137
   - env5test Test Requirements 137
   - env5test Options 138
   - env5test Test Modes 140
   - env5test Command-Line Syntax 140

23. **Frame Buffer Test (fbtest)** 141
   - fbtest Options 141
   - fbtest Test Modes 143
   - fbtest Command-Line Syntax 143

24. **Fast Frame Buffer Test (ffbtest)** 145
   - ffbtest Test Requirements 145
   - ffbtest Options 146
   - ffbtest Test Modes 152
   - ffbtest Command-Line Syntax 153

25. **Floating Point Unit Test (fputest)** 155
   - fputest Subtests 155
fputest Options 156
fputest Test Modes 158
fputest Command-Line Syntax 158

26. IEEE 1394 Camera Test (fwcamtest) 159
fwcamtest Test Requirements 159
    Start a Window Environment 159
    Testing Through a Remote Connection 160
fwcamtest Subtests 160
fwcamtest Options 160
fwcamtest Test Modes 162
fwcamtest Command-Line Syntax 162

27. Graphics Frame Buffer Test (gfbtest) 163
gfbtest Test Requirements 164
gfbtest Options 164
gfbtest Test Modes 171
gfbtest Command Line Syntax 171

28. PGX32 Frame Buffer Test (gfxtest) 175
gfxtest Test Requirements 175
gfxtest Options 176
gfxtest Test Modes 178
gfxtest Command-Line Syntax 178

29. I2C Bus Test (i2ctest) 179
i2ctest Test Requirements 179
i2ctest Options 180
i2ctest Test Modes 181
i2ctest Command-Line Syntax 181
30. **Expert3D Frame Buffer Test (ifbtest)** 183
   - ifbtest Test Requirements 183
     - Preparation for ifbtest 184
   - ifbtest Options 185
   - ifbtest Test Modes 189
   - ifbtest Command-Line Syntax 189

31. **Intelligent Fibre Channel Processor Test (ifptest)** 191
   - ifptest Subtests 191
   - ifptest Options 192
   - ifptest Test Modes 194
   - ifptest Command-Line Syntax 195

32. **Level 1 Data Cache Test (l1dcachetest)** 197
   - l1dcachetest Options 197
   - l1dcachetest Test Modes 199
   - l1dcachetest Command-Line Syntax 199

33. **Level 2 Cache Test (l2dcachetest)** 201
   - l2dcachetest Options 201
   - l2dcachetest Test Modes 203
   - l2dcachetest Command-Line Syntax 203

34. **LOMlite Alarm Test (lomlitetest)** 205
   - lomlitetest Requirements 205
   - lomlitetest Subtests 206
   - lomlitetest Options 207
   - lomlitetest Test Modes 208
   - lomlitetest Command-Line Syntax 208

35. **M64 Video Board Test (m64test)** 209
36. Cache Consistency Test (mpconstest) 215
   mpconstest Test Requirements 216
   mpconstest Subtests 217
   mpconstest Options 218
   mpconstest Test Modes 221
   mpconstest Command-Line Syntax 221

37. Multiprocessor Test (mptest) 223
   mptest Options 223
   mptest Test Modes 227
   mptest Command-Line Syntax 227

38. Network Hardware Test (nettest) 229
   nettest Options 230
   nettest Test Modes 232
   nettest Command-Line Syntax 233

39. Ethernet Loopback Test (netlbtest) 235
   netlbtest Test Requirements 235
   netlbtest Options 236
   netlbtest Test Modes 238
   netlbtest Command-Line Syntax 238

40. PCMCIA Modem Card Test (pcsertest) 241
   pcsertest Options 241
   pcsertest Test Mode 243
   pcsertest Command-Line Syntax 243
41. SPARCstorage Array Controller Test (plntest) 245
   plntest Controller Test 245
   Probing for SSA Controller Devices 246
   plntest Options 247
   plntest Test Modes 249
   plntest Command-Line Syntax 249

42. Physical Memory Test (pmemtest) 251
   pmemtest Options 251
   pmemtest Test Modes 254
   pmemtest Command-Line Syntax 254

43. Qlogic 2202 Board Test (qlctest) 257
   qlctest Subtests 257
   qlctest Options 258
   qlctest Test Modes 262
   qlctest Command-Line Syntax 262

44. Remote System Control (rsctest) 265
   rsctest Subtests 265
   rsctest Options 267
   rsctest Test Modes 270
   rsctest Command-Line Syntax 270

45. Serial Asynchronous Interface (PCI) Test (saiptest) 273
   saiptest Hardware Requirements 273
   saiptest Options 274
   saiptest Test Modes 277
   saiptest Command-Line Syntax 277
46. **Sun Enterprise Cluster 2.0 Network Hardware Test (scitest)** 281
   - scitest Options 281
   - scitest Test Modes 283
   - scitest Command-Line Syntax 283

47. **Smart Card Test (sctest)** 285
   - sctest Test Requirements 285
   - sctest Subtests 287
   - sctest Options 287
   - sctest Test Modes 289
   - sctest Command-Line Syntax 289

48. **Environmental Sensing Card Test (sentest)** 291
   - sentest Options 291
   - sentest Test Modes 293
   - sentest Command-Line Syntax 294

49. **Soc+ Host Adapter Card Test (socaltest)** 295
   - socaltest Options 295
   - socaltest Test Modes 298
   - socaltest Command-Line Syntax 298

50. **Serial Parallel Controller Test (spiftest)** 299
   - spiftest Hardware Requirements 299
   - spiftest Options 300
   - spiftest Test Modes 303
   - spiftest Command-Line Syntax 303

51. **Serial Ports Test (sptest)** 305
   - sptest Synchronous Testing Software Requirements 307
   - sptest Options 307
52. **SunHSI Board Test** *(sunlink)* 315
   - sunlink Test Requirements 315
   - sunlink Options 316
   - sunlink Loopback Connectors 317
   - sunlink Test Modes 318
   - sunlink Command-Line Syntax 318

53. **SunPCI II Test** *(sunpci2test)* 321
   - sunpci2test Test Requirements 321
   - sunpci2test Options 322
   - sunpci2test Test Modes 323
   - sunpci2test Command-Line Syntax 323

54. **SuperI/O Test** *(sutest)* 325
   - Loopback Connectors 325
   - sutest Options 326
   - sutest Test Modes 328
   - sutest Command-Line Syntax 328

55. **System Test** *(systest)* 331
   - systest Options 331
   - systest Test Modes 333
   - systest Command-Line Syntax 333

56. **Tape Drive Test** *(tapetest)* 335
   - tapetest Test Requirements 335
   - tapetest Options 335
   - tapetest Test Modes 339
25-pin Port A-to-A Port B-to-B Loopback Plug 370
96-Pin Female Loopback Connector 371
96-Pin Female Special Loopback Connector 373
37-Pin RS-449 Loopback Cable 374
37-Pin RS-449 Loopback Plug 375
9-pin Male Single-Port Loopback Plug 376
9-pin Female Single-Port Loopback Plug 376
9-Pin to 25-Pin Port-to-Port Loopback Cable 377
9-Pin to 9-Pin Port-to-Port Loopback Cable 378
NT to TE Loopback Cable 378
Twisted-Pair Ethernet (TPE) Loopback Cable For Fast Ethernet 379
TPE Loopback Cable For Gigabit Ethernet 379
Preface

SunVTS™ 5.0 is the Sun Microsystems™ Validation Test Suite. SunVTS is a comprehensive software diagnostic package that tests and validates Sun™ hardware by verifying the configuration and functionality of most hardware controllers, devices, and platforms.

SunVTS is primarily used from a graphical user interface (GUI), for the Common Desktop Environment (CDE). This book describes SunVTS tests that run on machines with SPARC™ architectures. The descriptions include specific test options, procedures, and error messages.

This book is primarily written as a reference for SunVTS test specific information. Refer to the SunVTS 5.0 User’s Guide for overall SunVTS information. Developers or experienced users who want to run the SunVTS diagnostic application will find these documents useful.

Before You Read This Book

In order to make full use of the information in this document, you may need access to the following documents:

- SunVTS 5.0 User’s Guide
- SunVTS Quick Reference Card
How This Book Is Organized

This book is organized as follows:

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

The remaining chapters describe the individual SunVTS tests, their options, applicable test modes, and command-line syntax. These chapters are arranged in alphabetical order according to each 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 does not contain information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices.

See one or more of the following for this information:

- Solaris Handbook for Sun Peripherals
- Online documentation for the Solaris™ software environment at docs.sun.com
- Other software documentation that you received with your system
## Typographic Conventions

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

## Shell Prompts

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>C shell</td>
<td>machine-name%</td>
</tr>
<tr>
<td>C shell superuser</td>
<td>machine-name#</td>
</tr>
<tr>
<td>Bourne shell and Korn shell</td>
<td>$</td>
</tr>
<tr>
<td>Bourne shell and Korn shell superuser</td>
<td>#</td>
</tr>
</tbody>
</table>
Related Documentation

This manual covers SunVTS revision 5.0 tests. It serves as a reference companion to the SunVTS 5.0 manuals listed below.

<table>
<thead>
<tr>
<th>Application</th>
<th>Title</th>
<th>Part Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation and Navigation</td>
<td>SunVTS 5.0 User’s Guide</td>
<td>816-1666-10</td>
</tr>
<tr>
<td>Quick Reference Service</td>
<td>SunVTS Quick Reference</td>
<td>816-0861-10</td>
</tr>
</tbody>
</table>

Accessing Sun Documentation Online

A broad selection of Sun system documentation is located at:

http://www.sun.com/products-n-solutions/hardware/docs

A complete set of Solaris documentation and many other titles are located at:

http://docs.sun.com

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

docfeedback@sun.com

Please include the part number (816-1667-10) of your document in the subject line of your email.
CHAPTER 1

Introduction

This manual describes SunVTSTM Version 5.0 tests that are distributed on the Sun Computer Systems Supplement CD.

The SunSTM 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.

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 SolarisTM 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. This book covers the individual test options and requirements. For overall test configuration modes and options refer to the SunVTS 5.0 User’s Guide.

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 5.0 User’s Guide for installation information.

Test Requirements

SunVTS version 5.0 was first introduced and designed to run in the Solaris 9 operating environment. It is recommended that you run SunVTS 5.0 in the Solaris 9 operating environment.

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.

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/sparcv9/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 5.0 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 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 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 16). TABLE 1-1 describes the various SunVTS user interfaces. Refer to the *SunVTS 5.0 User’s Guide* for more information on these interfaces.

**TABLE 1-1**  
SunVTS System Interfaces

<table>
<thead>
<tr>
<th>SunVTS System Interfaces</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphical user interfaces (GUIs)</td>
<td>Users can select tests and test options by pointing and clicking with a mouse button in the CDE interface.</td>
</tr>
<tr>
<td>TTY interface</td>
<td>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.</td>
</tr>
<tr>
<td>Command-line execution</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Note** – To increase or decrease a numeric value in a SunVTS 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—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 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 (CDE)

TABLE 1-2 Test Parameter Options Dialog Box Items

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>Information such as device type, capacity, revision, and serial numbers for the selected device. This information cannot be changed.</td>
</tr>
<tr>
<td>Options</td>
<td>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.</td>
</tr>
<tr>
<td>Within Instance</td>
<td>Provides the means to apply the settings:</td>
</tr>
<tr>
<td></td>
<td>• to this device only with Apply, or</td>
</tr>
<tr>
<td></td>
<td>• to all devices within this group with Apply to Group, or</td>
</tr>
<tr>
<td></td>
<td>• to all devices (of the same device type for all controllers) with Apply to All.</td>
</tr>
<tr>
<td></td>
<td>The option settings are only applied to one instance of the test.</td>
</tr>
</tbody>
</table>
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 “32-Bit and 64-Bit Tests” on page 13.

Standard Command-Line Arguments

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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-s</td>
<td>Runs a test as though it were invoked from the SunVTS kernel (vtsk). The default is to send the output to stdout or stderr.</td>
</tr>
<tr>
<td>-c</td>
<td>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.</td>
</tr>
<tr>
<td>-r</td>
<td>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.</td>
</tr>
<tr>
<td>-u</td>
<td>Displays command-line usage information.</td>
</tr>
<tr>
<td>-v</td>
<td>Runs the test in Verbose mode and displays messages with more detailed information about the testing process. The default is false.</td>
</tr>
<tr>
<td>-d</td>
<td>Runs the test in debug mode and displays messages to help programmers debug their test code. The default is false.</td>
</tr>
<tr>
<td>-t</td>
<td>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.</td>
</tr>
<tr>
<td>-e</td>
<td>Runs the test in Stress mode by increasing the system load. The default is false.</td>
</tr>
<tr>
<td>-l</td>
<td>Runs the test in Online Functional mode. This is the same mode that tests run in when executed with the <code>vtsui.online</code> command. It is a non-intrusive version that will not significantly affect other applications. See the note below. The default is true.</td>
</tr>
<tr>
<td>-n</td>
<td>Runs the test in Connection mode. See the note below. The default is false.</td>
</tr>
<tr>
<td>-f</td>
<td>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.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| -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). |

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 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=cgthree0,lock=enable
```

(See the test command line argument descriptions in this manual.)

---

**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 Common Desktop Environment: User’s Guide in the Solaris 9 User Collection. This document is available at: http://www.docs.sun.com.

---

**Caution** – If you are using the CDE 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). 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.
Advanced Frame Buffer Test  
(afbtest)

The afbtest verifies the functionality of the advanced frame buffer (AFB).

The afbtest can detect and adapt to the various video modes of the AFB. Instead of only running in one standard graphics mode, all tests can run in any mode. In stereo mode, all tests write into the right and left eyes unless you specify otherwise.

You can interrupt afbtest using Control-c.

Test accuracy is checked using a checksum algorithm. Possible locations of failing pixels are colored chartreuse to help visually identify their position.

Caution – Do not run any other application or screen saver program that uses the AFB accelerator port while running afbtest. This combination causes SunVTS to return incorrect errors.

afbtest Test Requirements

Disable all screen savers before testing any graphics device. Type `xset s off` at a UNIX® prompt to disable the Solaris screen saver.

For full instructions on testing frame buffers, please see “Testing Frame Buffers” on page 18.

afbtest requires approximately 29 MB of disk space in the /tmp directory to extract its working files. If this space is not available, the diagnostic will fail and report warning and error messages, indicating a lack of disk space.
afbtest Options

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

By default, all afbtest options are enabled.
FIGURE 2-1  afbtest Test Parameter Options Dialog Box
<table>
<thead>
<tr>
<th><strong>afgtest Options</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>3DRAM test</td>
<td>The 3DRAM test thoroughly tests the video memory in the AFB using 512-bit reads and writes. 3DRAM makes a full-screen pass, consisting of a write and a read to each pixel location, for each access mode on the list below. The data used can be either random or specified by the user. A second pass is made with the one’s complement of the data used in the first pass so that each memory location is tested with both a zero and a one. Errors in this subtest are attributes to the 3DRAM. A failing chip is indicated by (X, Y) locations and device-specific “U” numbers.</td>
</tr>
<tr>
<td>• DFB8R, DFB8G, DFB8B, DFB8X—Buffer A</td>
<td></td>
</tr>
<tr>
<td>• DFB24—Buffer A</td>
<td></td>
</tr>
<tr>
<td>• DFB32—Buffer A</td>
<td></td>
</tr>
<tr>
<td>• SFB8R, SFB8G, SFB8B, SFB8X—Buffer A</td>
<td></td>
</tr>
<tr>
<td>• SFB8R, SFB8G, SFB8B, SFB8X—Buffer B</td>
<td></td>
</tr>
<tr>
<td>• SFB32—Buffer A</td>
<td></td>
</tr>
<tr>
<td>• SFB32—Buffer B</td>
<td></td>
</tr>
<tr>
<td>• SFB32—Buffer C</td>
<td></td>
</tr>
<tr>
<td>• SFB64—Buffers A and C</td>
<td></td>
</tr>
<tr>
<td>• SFB64—Buffers B and C</td>
<td></td>
</tr>
</tbody>
</table>
3DRAM Logic test

The 3DRAM Logic test provides logical functionality to the AFB. The following services are tested:

- Compare Controls—Match AB
- Compare Controls—Magnitude AB
- Compare Controls—Match C
- Compare Controls—Magnitude C
- Match Mask—AB
- Magnitude Mask—AB
- Match Mask—C
- Magnitude Mask—C
- Raster Operations—RGB
- Raster Operations—X
- Raster Operations—YZ
- Plane Mask—RGB
- Plane Mask—X
- Plane Mask—Y
- Plane Mask—Z
- Group Enable—R, G, B, X
- Group Enable—Y, Z

Each function is tested separately with a series of SFB64 writes. A total of 16 writes are made for each different test case with Y coordinate values varying from 0 to 30 in increments of 2 pixels. This dotted column organization provides page thrashing and block flashing in all screen resolutions. For each operation all possible combinations are tested. For example, in ROP_RGB new==old there are three possible values: new < old, new == old, and new > old. Each of these cases are tested.

Five passes of the functions are made. Each pass writes into a different AFB address space: SFB32-A, SFB32-B, SFB32-C, SFB64-AC, and SFB64-BC. Note that the passes that write into the SFB32 address spaces are writing two pixels at a time because the tests use SFB64 writes.

Care is taken to ensure that all 3DRAM chips are tested. Errors in this subtest are attributed to the 3DRAM.
RAMDAC test

RAMDAC registers are tested using simple read/write patterns to determine if there are any bad bits. This includes all LUTs (4 CLUTs, PWLUT and OWLUT). afbtest ensures that data is actually being read from the RAMDAC and not being supplied by the driver.

RAMDAC on AFB can be in SEP8 or Combined mode. RAMDAC test detects the RAMDAC mode and tests the RAMDAC output for that mode. The RAMDAC Signature Register captures the pixels going to the screen. This test determines that all of the different data paths within the RAMDAC are functioning properly.

The data pattern is designed so all the data paths are tested, that is, all CLUTs, PWLUTs, and OWLUTs. A cursor is also displayed on the screen.

Errors in this test are attributed to the RAMDAC.

Microcode test

Microcode test generates the checksum for the microcode of each enabled float and compares all the checksums for equality.

Errors in this test are attributed to the microcode PROMS & SRAMS.

Rendering Pipeline test

Rendering Pipeline test uses the rendering pipeline tests developed for the FFB stand-alone diagnostics. Each FFB primitive is tested thoroughly with a variety of sources and configurations:
- Dots
- Anti-aliased dots
- Lines using all four line drawing primitives
- Triangles
- Polygons
- Rectangles
- Fonts

Errors in this test are attributed to the Draw Chips.

Fast Fill/Vertical Scroll test

Fast Fill/Vertical Scroll primitives are separated from the Rendering Pipeline tests because of their dependence on screen type. There are three different tests, one for each screen type. Each test uses both block and page mode fast fills.

Errors in this test are attributed to the Draw Chips.

<table>
<thead>
<tr>
<th>afbtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAMDAC test</td>
<td>RAMDAC registers are tested using simple read/write patterns to determine if there are any bad bits. This includes all LUTs (4 CLUTs, PWLUT and OWLUT). afbtest ensures that data is actually being read from the RAMDAC and not being supplied by the driver.</td>
</tr>
<tr>
<td>Microcode test</td>
<td>Microcode test generates the checksum for the microcode of each enabled float and compares all the checksums for equality.</td>
</tr>
<tr>
<td>Rendering Pipeline test</td>
<td>Rendering Pipeline test uses the rendering pipeline tests developed for the FFB stand-alone diagnostics. Each FFB primitive is tested thoroughly with a variety of sources and configurations: Dots, Anti-aliased dots, Lines using all four line drawing primitives, Triangles, Polygons, Rectangles, Fonts</td>
</tr>
<tr>
<td>Fast Fill/Vertical Scroll test</td>
<td>Fast Fill/Vertical Scroll primitives are separated from the Rendering Pipeline tests because of their dependence on screen type. There are three different tests, one for each screen type. Each test uses both block and page mode fast fills.</td>
</tr>
</tbody>
</table>
Pixel Process test  The Pixel Processor test, a subtest, exercises the options selected by the AFB’s Pixel Processor Control (PPC) register:
- Auxiliary clipping (additive and subtractive)
- Depth cueing
- Alpha blend
- Viewport clip (2D and 3D)
- Area pattern (transparent and opaque)

Errors in this test are attributed to the Draw Chips.

AFB Dots test  This test uses the AFB primitive tests developed for the AFB stand-alone diagnostics. AFB Dots are tested thoroughly with a variety of sources and configurations:
- Dots
- Anti-aliased dots
- Big dots

Errors in this test are attributed to the Command & Draw Chips.

AFB Lines test  This test uses the AFB primitive tests developed for the AFB stand-alone diagnostics. AFB Lines are tested thoroughly with a variety of sources and configurations:
- Jaggy lines
- Anti-aliased lines
- Lines with patterns
- Bresenham lines
- Wide lines drawn as lines and triangles

Errors in this test are attributed to the Command & Draw Chips.

AFB Triangles test  This test uses the AFB primitive tests developed for the AFB stand-alone diagnostics. AFB Triangles are tested thoroughly with a variety of sources and configurations:
- Triangles drawn clockwise & counter clockwise
- Triangles drawn as stripes
- Independent triangles
- Triangles drawn as stars
- Triangles with facet normals

Errors in this test are attributed to the Command & Draw Chips.

<table>
<thead>
<tr>
<th>afbtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel Process test</td>
<td>The Pixel Processor test, a subtest, exercises the options selected by the AFB’s Pixel Processor Control (PPC) register:</td>
</tr>
<tr>
<td></td>
<td>• Auxiliary clipping (additive and subtractive)</td>
</tr>
<tr>
<td></td>
<td>• Depth cueing</td>
</tr>
<tr>
<td></td>
<td>• Alpha blend</td>
</tr>
<tr>
<td></td>
<td>• Viewport clip (2D and 3D)</td>
</tr>
<tr>
<td></td>
<td>• Area pattern (transparent and opaque)</td>
</tr>
<tr>
<td></td>
<td>Errors in this test are attributed to the Draw Chips.</td>
</tr>
<tr>
<td>AFB Dots test</td>
<td>This test uses the AFB primitive tests developed for the AFB stand-alone diagnostics. AFB Dots are tested thoroughly with a variety of sources and configurations:</td>
</tr>
<tr>
<td></td>
<td>• Dots</td>
</tr>
<tr>
<td></td>
<td>• Anti-aliased dots</td>
</tr>
<tr>
<td></td>
<td>• Big dots</td>
</tr>
<tr>
<td></td>
<td>Errors in this test are attributed to the Command &amp; Draw Chips.</td>
</tr>
<tr>
<td>AFB Lines test</td>
<td>This test uses the AFB primitive tests developed for the AFB stand-alone diagnostics. AFB Lines are tested thoroughly with a variety of sources and configurations:</td>
</tr>
<tr>
<td></td>
<td>• Jaggy lines</td>
</tr>
<tr>
<td></td>
<td>• Anti-aliased lines</td>
</tr>
<tr>
<td></td>
<td>• Lines with patterns</td>
</tr>
<tr>
<td></td>
<td>• Bresenham lines</td>
</tr>
<tr>
<td></td>
<td>• Wide lines drawn as lines and triangles</td>
</tr>
<tr>
<td></td>
<td>Errors in this test are attributed to the Command &amp; Draw Chips.</td>
</tr>
<tr>
<td>AFB Triangles test</td>
<td>This test uses the AFB primitive tests developed for the AFB stand-alone diagnostics. AFB Triangles are tested thoroughly with a variety of sources and configurations:</td>
</tr>
<tr>
<td></td>
<td>• Triangles drawn clockwise &amp; counter clockwise</td>
</tr>
<tr>
<td></td>
<td>• Triangles drawn as stripes</td>
</tr>
<tr>
<td></td>
<td>• Independent triangles</td>
</tr>
<tr>
<td></td>
<td>• Triangles drawn as stars</td>
</tr>
<tr>
<td></td>
<td>• Triangles with facet normals</td>
</tr>
<tr>
<td></td>
<td>Errors in this test are attributed to the Command &amp; Draw Chips.</td>
</tr>
</tbody>
</table>
Lighting test

The Lighting test exercises AFB float and lighting microcode. This test lights an object with maximum number of lights (32) that AFB can handle in hardware. A checksum is generated for the rendered image and compared with the checksum generated for the same image on a known good system.

Errors in this test are attributed to the Float & Microcode SRAMs.

Texture Processor test

The Texture Processor test exercises the different options of the AFB’s Texture Pixel Processor Control (TPPC) register:

• Texture Minification
• Texture Magnification
• Blend
• Decal
• Modulation

Errors in this test are attributed to the Draw Chips.

AFB Mix test

The AFB Mix test draws different primitives with variety combinations of sources and configurations, exercising all the Draw, Float, Microcode, and 3DRAM chips on AFB. This test is to stress the AFB.

Errors in this test are attributed to Draw, Float, Microcode, and/or 3DRAM Chips.

Picking test

The Picking test exercises the pick detect login of the 3DRAM. We define a pick detect window and make sure that writes to the window are picked, and writes outside the window are not picked. The test is repeated once for each 3DRAM.

Errors in this test are attributed to the 3DRAM.

Arbitration test

The Arbitration test, a subtest, continuously renders an object into the accelerator port while doing reads and writes through the direct port. A picture is rendered into all 32 planes of the B buffer while the other process does 32-bit DFB reads and writes in the A plane. This subtest simulates conditions in the real world, where rendering processes and windows operations run concurrently.

Errors in this test are attributed to the Context switching between DFB and SFB.

<table>
<thead>
<tr>
<th>afbtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lighting test</td>
<td>The Lighting test exercises AFB float and lighting microcode. This test lights an object with maximum number of lights (32) that AFB can handle in hardware. A checksum is generated for the rendered image and compared with the checksum generated for the same image on a known good system. Errors in this test are attributed to the Float &amp; Microcode SRAMs.</td>
</tr>
<tr>
<td>Texture Processor test</td>
<td>The Texture Processor test exercises the different options of the AFB’s Texture Pixel Processor Control (TPPC) register: • Texture Minification • Texture Magnification • Blend • Decal • Modulation Errors in this test are attributed to the Draw Chips.</td>
</tr>
<tr>
<td>AFB Mix test</td>
<td>The AFB Mix test draws different primitives with variety combinations of sources and configurations, exercising all the Draw, Float, Microcode, and 3DRAM chips on AFB. This test is to stress the AFB. Errors in this test are attributed to Draw, Float, Microcode, and/or 3DRAM Chips.</td>
</tr>
<tr>
<td>Picking test</td>
<td>The Picking test exercises the pick detect login of the 3DRAM. We define a pick detect window and make sure that writes to the window are picked, and writes outside the window are not picked. The test is repeated once for each 3DRAM. Errors in this test are attributed to the 3DRAM.</td>
</tr>
<tr>
<td>Arbitration test</td>
<td>The Arbitration test, a subtest, continuously renders an object into the accelerator port while doing reads and writes through the direct port. A picture is rendered into all 32 planes of the B buffer while the other process does 32-bit DFB reads and writes in the A plane. This subtest simulates conditions in the real world, where rendering processes and windows operations run concurrently. Errors in this test are attributed to the Context switching between DFB and SFB.</td>
</tr>
</tbody>
</table>
Stereo test displays an object in stereo mode with different images for the right and left eye. The user can verify proper operation by looking at the screen with stereo glasses and following the instructions being displayed. If the monitor type is not 1280x1024 at 76MHz, this test prints a warning message and does not execute.

To prevent this message from being displayed or written to the SunVTS information log, disable the stereo test in the Test Parameter Options dialog box. Only Sony P4 and N2 monitors support stereo resolutions. This test temporarily switches the monitor into stereo mode, renders a stereo image, performs a signature analysis on the stereo image (using the RAMDAC signature capture register), and after displaying the image for five seconds, restores the monitor to its previous resolution.

Errors in this test are attributed to the RAMDAC.

UART test The UART test tests both UART0 and UART1. First, UART memory is tested using simple read/write patterns to determine if there are any bad bits. Then data is written to UART 0/1 and the written data is read using the internal loopback in polling mode. The read data is verified with written data.

Errors in this test are attributed to UART and its SRAM memory chip.

### afbtest Test Modes

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

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full set of tests.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**TABLE 2-1 afbtest Options**

<table>
<thead>
<tr>
<th>afbtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stereo test</td>
<td>Stereo test displays an object in stereo mode with different images for the right and left eye. The user can verify proper operation by looking at the screen with stereo glasses and following the instructions being displayed. If the monitor type is not 1280x1024 at 76MHz, this test prints a warning message and does not execute.</td>
</tr>
</tbody>
</table>

To prevent this message from being displayed or written to the SunVTS information log, disable the stereo test in the Test Parameter Options dialog box. Only Sony P4 and N2 monitors support stereo resolutions. This test temporarily switches the monitor into stereo mode, renders a stereo image, performs a signature analysis on the stereo image (using the RAMDAC signature capture register), and after displaying the image for five seconds, restores the monitor to its previous resolution.

Errors in this test are attributed to the RAMDAC.

| UART test       | The UART test tests both UART0 and UART1. First, UART memory is tested using simple read/write patterns to determine if there are any bad bits. Then data is written to UART 0/1 and the written data is read using the internal loopback in polling mode. The read data is verified with written data. |

Errors in this test are attributed to UART and its SRAM memory chip. |
# afbtest Command-Line Syntax

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

## TABLE 2-3  afbtest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td><code>device_name</code> is the relative path name of the device being tested with respect to <code>/dev/fbs</code>; the default is <code>afb0</code>.</td>
</tr>
<tr>
<td>S=subtest_number</td>
<td><code>subtest_number</code> is the test number of the subtest to be run. Select from the subtests below. You can run multiple subtests by adding the subtest numbers together. For example, <code>n=0x3</code> runs both test 1 and test 2; <code>n=0x180</code> runs both test 0x080 and test 0x0100. You do not need the leading zeros.</td>
</tr>
<tr>
<td></td>
<td>• n—0x00001 3DRAM</td>
</tr>
<tr>
<td></td>
<td>• n—0x00002 3DRAM Logic</td>
</tr>
<tr>
<td></td>
<td>• n—0x00004 RAMDAC</td>
</tr>
<tr>
<td></td>
<td>• n—0x00008 Micro code</td>
</tr>
<tr>
<td></td>
<td>• n—0x00010 Rendering Pipeline</td>
</tr>
<tr>
<td></td>
<td>• n—0x00020 FastFill/Vertical Scroll</td>
</tr>
<tr>
<td></td>
<td>• n—0x00040 Pixel Processor</td>
</tr>
<tr>
<td></td>
<td>• n—0x00080 AFB Dots</td>
</tr>
<tr>
<td></td>
<td>• n—0x00100 AFB Lines</td>
</tr>
<tr>
<td></td>
<td>• n—0x00200 AFB Triangles</td>
</tr>
<tr>
<td></td>
<td>• n—0x00400 Lighting</td>
</tr>
<tr>
<td></td>
<td>• n—0x00800 Texture Processor</td>
</tr>
<tr>
<td></td>
<td>• n—0x02000 AFB Mix Test</td>
</tr>
<tr>
<td></td>
<td>• n—0x04000 Picking</td>
</tr>
<tr>
<td></td>
<td>• n—0x08000 Arbitration</td>
</tr>
<tr>
<td></td>
<td>• n—0x10000 Stereo</td>
</tr>
<tr>
<td></td>
<td>• n—0x40000 UART</td>
</tr>
</tbody>
</table>

- `-F #_of_subtest_loops` The number of times to repeat each subtest. The default is 1.

- `-B #_of_test_loops` The number of times to repeat a test loop before passing. The default is 1.

- `-P test_pattern` The test pattern number. The default is 0, for random patterns. You may also choose 0 for 0x0000000, 3 for 0x3333333, 5 for 0x5555555, or 9 for 0x9999999.
Note – 64-bit tests are located in the sparcv9 subdirectory: 
/opt/SUNWvts/bin/sparcv9/testname. If the test is not present in this directory, 
then it may only be available as a 32-bit test. For more information refer to “32-Bit 
and 64-Bit Tests” on page 13.

Note – Errors returned by afbtest are nonspecific: It is not possible to determine 
which component caused a failure. In all error conditions, the field replaceable unit 
(FRU) is the entire AFB.
CHAPTER 3

Alarm Card Test (alarmtest)

The alarmtest exercises the Alarm Card on the Sun Netra™ CT 400 and CT 800 systems.

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

This test is not scalable.

Note – Do not run alarmtest and rsctest at the same time. Tests may return incorrect results.

Note – The Netra CT 400/800 system only runs the 64-bit OS (to take full advantage of UltraSPARCII). Although, alarmtest is available in 32-bit and 64-bit mode, only the 64-bit version of alarmtest is run on a Netra CT 400/800 system.

alarmtest Subtests

alarmtest consists of seven subtests which test and report on the following:

- Ethernet
- Serial ports
- PC-card (PCMCIA) socket
- Flash memory
- SEEPROM
- TOD
- Alarmport
alarmtest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 3-1  alarmtest Test Parameter Options Dialog Box
## TABLE 3-1  alarmtest Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enet_Test</td>
<td>Enables or disables Ethernet testing.</td>
</tr>
<tr>
<td>Edata_Pattern_Type</td>
<td>Selects the type of data pattern for Enet_Test: Sequential, Random, or both.</td>
</tr>
<tr>
<td>Num_Packets</td>
<td>Specifies the number of data packets to be sent in one test loop.</td>
</tr>
<tr>
<td>Target_IP_Addr</td>
<td>Specifies the IP address of a host to use for the ping test.</td>
</tr>
<tr>
<td>Etest_Type</td>
<td>Selects any or all internal, external, Phy (ethernet transceiver), or ping tests.</td>
</tr>
<tr>
<td>Serial_Test</td>
<td>Enables or disables serial_test.</td>
</tr>
<tr>
<td>Loopback_Type</td>
<td>Selects internal loopback, external loopback, or both.</td>
</tr>
<tr>
<td>Sdata_Pattern_Type</td>
<td>Selects the type of data pattern for serial_test: Sequential, Random, or both.</td>
</tr>
<tr>
<td>STest_Type</td>
<td>Selects ports to be tested: c, d, u, or v.</td>
</tr>
<tr>
<td>TTYU_Baud</td>
<td>Selects the alarm card’s COM1 port baud rate.</td>
</tr>
<tr>
<td>TTYV_Baud</td>
<td>Selects the alarm card’s COM2 port baud rate.</td>
</tr>
<tr>
<td>PC_Card_Test</td>
<td>Enables or disables PC card (PCMCIA) testing.</td>
</tr>
<tr>
<td>PC_Card_Type</td>
<td>Specifies the card type for the PC Card: modem or serial i/o.</td>
</tr>
<tr>
<td>Flash_Test</td>
<td>Enables or disables the flash checksum test.</td>
</tr>
<tr>
<td>SEEPRM_test</td>
<td>Enables or disables the SEEPRM checksum test.</td>
</tr>
<tr>
<td>TOD_test</td>
<td>Enables or disables the TOD checksum test.</td>
</tr>
<tr>
<td>ALARMPORT</td>
<td>Enables or disables the alarmport test.</td>
</tr>
<tr>
<td>ALARMNUM</td>
<td>Selects any or all alarm ports to be tested: 0, 1, 2, 3.</td>
</tr>
<tr>
<td>ALARM0ON</td>
<td>Turns on, turns off, or toggles (on then off) alarm port 0.</td>
</tr>
<tr>
<td>ALARM1ON</td>
<td>Turns on, turns off, or toggles (on then off) alarm port 1.</td>
</tr>
<tr>
<td>ALARM2ON</td>
<td>Turns on, turns off, or toggles (on then off) alarm port 2.</td>
</tr>
<tr>
<td>ALARM3ON</td>
<td>Turns on, turns off, or toggles (on then off) alarm port 3.</td>
</tr>
</tbody>
</table>

**Note** – The alarmport test requires a visual check using an LED box.
alarmtest Loopbacks

The loopback tests use the following external loopbacks:

- Ethernet loopback test—standard RJ-45 connector. Connect pin 1 to pin 3, and pin 2 to pin 6.
- Serial loopback test for Netra ct 800—DB-9 connector. Connect pin 2 to pin 3, pins 4 and 6 to pin 1, and pin 7 to pin 8.
- Serial loopback test for Netra ct 400—RJ-45. Connect pin 6 to pin 3, pin 1 to pin 8, and pin 2 to pin 7.

alarmtest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Reports the status of the alarm card.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the subtests for Ethernet, Serial, Flash, PCMCIA, SEEPROM, TOD, and Alarmport.</td>
</tr>
</tbody>
</table>

alarmtest Command-Line Syntax

```
/opt/SUNWvts/bin/alarmtest standard_arguments -o enet=E(nable)/D(isable), epatttype=seq+rand, epkts=number_packets, target=IP_address, etest=I+E+H+P, serial=E(nable)/D(isable), slb=I+E, spatttype=seq+rand, stest=c+d+u+v, pccard=E(nable)/D(isable), pccardtype=modem/serial, flash=E(nable)/D(isable), seeeprom=E(nable)/D(isable), tod=E(nable)/D(isable), ttyubaud=|
```
**TABLE 3-3** alarmtest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>enet=ENABLE/DISABLE</td>
<td>Enables or disables Ethernet testing.</td>
</tr>
<tr>
<td>epattype=SEQ+RAND</td>
<td>Selects the type of data pattern for Enet_Test: Sequential, Random, or both.</td>
</tr>
<tr>
<td>epkts=NUMBER_PACKETS</td>
<td>Specifies the number of data packets to be sent in one test loop.</td>
</tr>
<tr>
<td>target=IP_ADDRESS</td>
<td>Specifies the IP address of a host to use for the ping test.</td>
</tr>
<tr>
<td>etest=I+E+H+P</td>
<td>Selects any or all internal, external, Phy, or ping tests.</td>
</tr>
<tr>
<td>serial=ENABLE/DISABLE</td>
<td>Enables or disables serial_test.</td>
</tr>
<tr>
<td>slb=I+E</td>
<td>Selects internal loopback, external loopback, or both.</td>
</tr>
<tr>
<td>spattype=SEQ+RAND</td>
<td>Selects the type of data pattern for serial_test: Sequential, Random, or both.</td>
</tr>
<tr>
<td>stest=c+d+u+v</td>
<td>Selects ports to be tested: c, d, u, or v.</td>
</tr>
<tr>
<td>pccard=ENABLE/DISABLE</td>
<td>Enables or disables PC card (PCMCIA) testing.</td>
</tr>
<tr>
<td>pccardtype=MODEM/SERIAL</td>
<td>Specifies the card type for the PC Card: modem or serial i/o. Default is modem.</td>
</tr>
<tr>
<td>flash=ENABLE/DISABLE</td>
<td>Enables or disables the flash checksum test.</td>
</tr>
<tr>
<td>seeprom=ENABLE/DISABLE</td>
<td>Enables or disables the SEEPROM checksum test.</td>
</tr>
<tr>
<td>tod=ENABLE/DISABLE</td>
<td>Enables or disables the TOD checksum test.</td>
</tr>
<tr>
<td>ttyvbaud=ALL</td>
<td>SPECIFIC_BAUD</td>
</tr>
<tr>
<td>ttyvbaud=ALL</td>
<td>SPECIFIC_BAUD</td>
</tr>
<tr>
<td>aport=ENABLE/DISABLE</td>
<td>Enables or disables the alarmport test.</td>
</tr>
<tr>
<td>anum=0+1+2+3</td>
<td>Selects any or all alarm port to be tested: 0, 1, 2, 3</td>
</tr>
<tr>
<td>a0on=ON/OFF/TOGGLE</td>
<td>Turns on, turns off, or toggles (on then off) alarm port 0.</td>
</tr>
<tr>
<td>a1on=ON/OFF/TOGGLE</td>
<td>Turns on, turns off, or toggles (on then off) alarm port 1.</td>
</tr>
<tr>
<td>a2on=ON/OFF/TOGGLE</td>
<td>Turns on, turns off, or toggles (on then off) alarm port 2.</td>
</tr>
<tr>
<td>a3on=ON/OFF/TOGGLE</td>
<td>Turns on, turns off, or toggles (on then off) alarm port 3.</td>
</tr>
</tbody>
</table>
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 “32-Bit and 64-Bit Tests” on page 13.
SunATM Adapter Test (atmtest)

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

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

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

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

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

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

atmtest Test Requirements

atmtest can only be selected when the Intervention mode is enabled since it requires a loopback connector for external loopback testing. While Intervention mode is enabled, atmtest and nettest are both available as default selections; however, you must deselect nettest when testing the ATM device.
Bring the ATM interface down to make sure that the interface is in offline mode before running atmtest.

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

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

---

**atmtest Options**

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User’s Guide* for more details.
FIGURE 4-1 atmtest Test Parameter Options Dialog Box
<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>The post address, host ID, and domain name of the system being tested.</td>
</tr>
<tr>
<td>Total packets</td>
<td>The total number of packets sent. The default number of packets sent is 10000.</td>
</tr>
<tr>
<td>Number of VC</td>
<td>The default number of virtual circuits is 2. The <code>atmtest</code> uses these two virtual circuits to send out messages simultaneously. The message is received in sending order.</td>
</tr>
<tr>
<td>MAX_PKG_LEN</td>
<td>The maximum packet length to be used by the test to send out the data. The default number is 9140.</td>
</tr>
<tr>
<td>Outstanding_pkts</td>
<td>Describes the maximum number of outstanding packets. <code>atmtest</code> stops sending messages when the outstanding packet count is more than the number of packets this field specifies.</td>
</tr>
<tr>
<td>First_VC_no</td>
<td>Enables the user to set up the starting virtual circuit number to be used for each <code>atmtest</code> instance. <code>atmtest</code> can automatically avoid virtual circuit numbers that have already been used.</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>Enables the user to select different bandwidths to test. The default number is 14.</td>
</tr>
<tr>
<td>Loopback</td>
<td>Enables the user to select either the external loopback field or internal loopback field. The default selection is the external loopback field. A loopback connector is only needed for external loopback testing.</td>
</tr>
<tr>
<td>Print Warning</td>
<td>Disabled by default. Click Enable to see warning errors, such as retry on timeout.</td>
</tr>
</tbody>
</table>
### atmtest Test Modes

**TABLE 4-2** atmtest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests.</td>
</tr>
</tbody>
</table>

### atmtest Command-Line Syntax

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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device</td>
<td>Specifies the device name to be tested, such as ba0 or sa0.</td>
</tr>
<tr>
<td>tpkts=n</td>
<td>Specifies the number of packets to loopback, -1 for continuous [1...2147483647, -1].</td>
</tr>
<tr>
<td>nv=num_vc</td>
<td>Specifies the number of simultaneous virtual circuits to be tested.</td>
</tr>
<tr>
<td>ml=max_len</td>
<td>Specifies the maximum length of the random packet.</td>
</tr>
<tr>
<td>bw=bandwidth</td>
<td>Specifies the bandwidth in MBits/s of a virtual circuit.</td>
</tr>
<tr>
<td>opkts=n</td>
<td>Specifies the number of packets for each virtual circuit that can be transmitted without receiving a corresponding packet.</td>
</tr>
<tr>
<td>warn</td>
<td>When enabled, prints warning messages.</td>
</tr>
<tr>
<td>ld</td>
<td>The internal loopback mode is selected.</td>
</tr>
<tr>
<td>sd</td>
<td>Changes the payload data to static instead of random.</td>
</tr>
<tr>
<td>sl</td>
<td>Changes all packets to their maximum length.</td>
</tr>
<tr>
<td>nc</td>
<td>Instructs the test not to check the receive payload (improves throughput).</td>
</tr>
<tr>
<td>ns</td>
<td>Instructs the test not to exit on a packet reception failure.</td>
</tr>
<tr>
<td>vcf=n</td>
<td>Specifies the first virtual circuit number used.</td>
</tr>
</tbody>
</table>
**Note** – 64-bit tests are located in the `sparcv9` subdirectory: 
`/opt/SUNWvts/bin/sparcv9/testname`. If the test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to “32-Bit and 64-Bit Tests” on page 13.
Audio Test (*audiotest*)

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

This test will work with exclusive access devices (only one process or application available at a time), or with newer audio devices which support the software mixer feature available in the Solaris 8 operating environment.

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

This test is not scalable.

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

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

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

![FIGURE 5-1 audiotest Test Parameter Options Dialog Box](image)

**Note** – Upon startup, the SunVTS probe utility determines which audio implementation is present and adjusts the audiotest Option menu appropriately. Your dialog box may look different than the one pictured here, but will contain some or all of these options.
Note – The internal loopbacks are only active if the audio jacks are unused (nothing connected).

Some options can only be selected through the command line. See the command-line option descriptions in “audiotest Command-Line Syntax” on page 41.

TABLE 5-2  audiotest Options

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

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

#### TABLE 5-3 audiotest Test Modes

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

### audiotest Command-Line Syntax

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

#### TABLE 5-4 audiotest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=/dev/audio_device</td>
<td>Specifies the audio device to be tested. The default is dev=/dev/audio.</td>
</tr>
<tr>
<td>I=/dev/ioctl_device</td>
<td>Specifies the audio ioctl device to be tested. The default is /dev/ioctl.</td>
</tr>
<tr>
<td>M</td>
<td>Enables the Music Play test.</td>
</tr>
<tr>
<td>L</td>
<td>Enables the Loopback test.</td>
</tr>
<tr>
<td>Q</td>
<td>Enables the Quality test. This option does the same thing as L option except that it prints an extra status message upon completion.</td>
</tr>
<tr>
<td>S</td>
<td>Enables the speakerbox Controls test.</td>
</tr>
</tbody>
</table>
Specifies the type of Loopback test. The default is 1; the choices are listed below:

- 0—Codec Internal Loopback (CS4231 audio only)
- 1—Line-in/Line-out
- 2—Headphone/Line-in
- 3—Headphone/Microphone
- 4—Speaker/CD-input
- 11—Internal Line-in/Line-out
- 12—Internal Spk/Mic
- 13—Internal Headphone/Aux1
- 14—Internal Speaker/Aux1
- 15—Internal Headphone/Mic

Note: Test type 0 is always run by default on CS4231 audio implementations. Test types 3 and 4 require special hardware, and are used by manufacturing centers and special test facilities. Do not invoke these tests unless you have the required hardware.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T=\text{loopback_test_type})</td>
<td>Specifies the type of Loopback test. The default is 1; the choices are listed below:</td>
</tr>
<tr>
<td>X</td>
<td>Enables the Audio Crystal test.</td>
</tr>
<tr>
<td>E</td>
<td>Continues testing if an error occurs.</td>
</tr>
<tr>
<td>LE</td>
<td>Loops on error. This plays the signal data in a continuous loop.</td>
</tr>
<tr>
<td>(\text{CD})</td>
<td>Enables the \text{cdtest}. This is for systems with an internal CD-ROM drive. A CD-ROM with music tracks must be loaded prior to running this test.</td>
</tr>
<tr>
<td>(\text{CDD=CD_device_name})</td>
<td>Specifies the raw device name for the CD-ROM drive. The default is \text{CDD=/dev/rdsk/c0t6d0s0}.</td>
</tr>
<tr>
<td>(\text{CDT=number})</td>
<td>Specifies the track number of the CD-ROM to play. The default is to play the first track on the disc.</td>
</tr>
<tr>
<td>(\text{CDG=play_gain})</td>
<td>Specifies the play gain of the CD Play test (0 to 255). The default is 120.</td>
</tr>
<tr>
<td>(\text{CDL=play_time})</td>
<td>Specifies the number of seconds to run the CD Play test. The default is 30 seconds.</td>
</tr>
</tbody>
</table>
Note – 64-bit tests are located in the sparcv9 subdirectory: /opt/SUNWvts/bin/sparcv9/testname. If the test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to “32-Bit and 64-Bit Tests” on page 13.
Bidirectional Parallel Port Printer Test (bpptest)

The bpptest verifies the functionality of the bidirectional parallel port. SBus printer cards have two printer ports: one for any printer device and one for a parallel port printer.

The bpptest also verifies that your SBus card and its parallel port are working properly by attempting to transfer a data pattern from the SBus card to the printer.

Two indications show that the card and printer are functioning properly: First, you can see from the SunVTS Status window that bpptest made a successful pass, and second, that the pattern transmitted to the printer printed correctly.

If the bpptest passes successfully, you know that the SBus DMA circuitry, the printer, and the device driver are functioning properly.

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

bpptest Hardware and Software Requirements

The SBus printer card and device drivers must be installed to run bpptest. A printer must be connected to the bidirectional parallel port, and be powered-up. If both a SPARCprinter and a parallel port printer are connected to the SBus card, you can test both devices at the same time.
bpptest 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.

![bpptest Test Parameter Options Dialog Box](image)

FIGURE 6-1 bpptest Test Parameter Options Dialog Box
TABLE 6-1  bpptest Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>Determines the direction of data transfer; this field is informational only. <em>Writeonly</em> is the only option currently available. Data is transferred only from the SBus printer card to the printer.</td>
</tr>
</tbody>
</table>
| Mode   | Sets the print intervals. This option lets you select the intervals at which the test image is printed. The default setting is fast. In the online and Functional tests this option is set to medium. In the Functional test, the choices are:  
* fast—prints an image every 10 seconds.  
* medium—prints an image every 12 minutes.  
* extended—prints an image every 30 minutes. |

**bpptest Test Modes**

This test supports Connection and Functional test modes.

TABLE 6-2  bpptest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>In this mode, <em>bpptest</em> verifies that a bidirectional parallel port is configured on the system. The success of the <em>bpptest</em> in this mode indicates that the bidirectional parallel port hardware and the software driver are installed on the system.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>The testing done in this mode registers a failure if the port is found busy. This is because SunVTS tests make the assumption that all the resources will be available for testing in the Functional test and therefore the unavailability of the device is interpreted as an indication of a fault condition.</td>
</tr>
</tbody>
</table>
**bpptest Command-Line Syntax**

```
/opt/SUNWvts/bin/bpptest standard_arguments -o dev=device_name, access=writeonly | readonly, mode=mode
```

### TABLE 6-3  bpptest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dev=device_name</code></td>
<td>Specifies the name of the device. This should be of the form /dev/bpp#, where # is the minor number of the device.</td>
</tr>
<tr>
<td><code>access=writeonly</code></td>
<td>Determines test mode. writeonly is the only option currently available.</td>
</tr>
<tr>
<td><code>mode=mode</code></td>
<td>Sets the test image print rate. The test image is a continuous printout of the ASCII character set. Possible rates are:</td>
</tr>
<tr>
<td></td>
<td>• fast — prints the test image at 10-second intervals.</td>
</tr>
<tr>
<td></td>
<td>• medium — prints the test image at 12-minute intervals.</td>
</tr>
<tr>
<td></td>
<td>• extended — prints the test image at 30-minute intervals.</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Compact Disc Test (cdtest)

The cdtest checks the CD-ROM unit by reading the CD.

cdtest is not a scalable test.

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.

**Note** – Load a compact disc into the drive before starting the test. See the explanation of CD types in TABLE 7-1.

Volume Management and Compact Discs

cdtest tests the CD-ROM drive(s) even if the Volume Manager is not running. If the Volume Manager is running and no media is installed in the CD-ROM 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 a CD in the drive.
cdtest 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.

![cdtest Test Parameter Options Dialog Box](image)

---

** FIGURE 7-1  cdtest Test Parameter Options Dialog Box **
### TABLE 7-1  cdtest Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD Type</td>
<td>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, this option has a default value of other. Note: Your choice must correspond with the disc used for testing.</td>
</tr>
<tr>
<td>% Data/Track</td>
<td>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%.</td>
</tr>
<tr>
<td>Read Mode</td>
<td>cdtest 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.</td>
</tr>
<tr>
<td>Audio Test</td>
<td>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.</td>
</tr>
<tr>
<td>Volume</td>
<td>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.</td>
</tr>
</tbody>
</table>
**cdtest Test Modes**

This test supports Connection and Functional tests.

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>In this mode, cdtest verifies that a CD-ROM drive is connected to and configured in the system.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>This mode is similar to Functional online mode except that 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.</td>
</tr>
</tbody>
</table>

**cdtest Command-Line Syntax**

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=raw_device_name</td>
<td>Specifies the name of the raw device to be tested.</td>
</tr>
<tr>
<td>read=random</td>
<td>sequential</td>
</tr>
<tr>
<td>data=%_of_data</td>
<td>Sets the percentage of data to be tested. You can specify 0 to 100 percent.</td>
</tr>
<tr>
<td>vol=volume</td>
<td>Controls the audio volume. You can specify 0 through 255; the default is 255.</td>
</tr>
<tr>
<td>audio=enable</td>
<td>disable</td>
</tr>
<tr>
<td>type=CD_type</td>
<td>Specifies the type of CD used for the test. The choices are: pdo, multi-session, sunos and other; the default is other.</td>
</tr>
</tbody>
</table>
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 “32-Bit and 64-Bit Tests” on page 13.
Color Graphics Frame Buffer Test (cg14test)

cg14test checks the cg14 frame buffer card. cg14test is specific to the VSIMM (Video SIMM) devices in the SPARCstation 10 SX and the SPARCstation 20 SX.

Caution – Do not close the SunVTS window to an icon while it is running frame buffer tests.

cg14test Requirements

Ensure that the frame buffer locking option is enabled from the Options window.

For full instructions on testing frame buffers, see “Testing Frame Buffers” on page 18.

cg14test Groups

There are nine test groups in cg14test:

1. MDI and VBC Chip Control Registers
2. Memory Chips
3. MDI Chip Cursor Registers
4. MDI Chip CLUT Registers  
5. DAC Chip Registers  
6. MDI Chip XLU Registers  
7. CG14 Display (visual only)  
8. MDI Chip Testmode Readback in 8-bit mode  
9. Driver IOCTLs

**TABLE 8-1  cg14 Test Groups**

<table>
<thead>
<tr>
<th>Test Groups</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware (Groups 1-6)</td>
<td>These test groups are tested by opening <code>/dev/fbs/cgfourteenX</code>, mapping the MDI Control Address Space, modifying the target test location (using direct writes to the mapped address space), reading from the mmapped address space for verification, and closing the device.</td>
</tr>
<tr>
<td>Visual Pattern (Group 7)</td>
<td>This subtest loads a visual pattern of 256 colors, then rotates the pattern around. You verify this test by seeing it display.</td>
</tr>
</tbody>
</table>
| Data Propagation (Group 8)| This test group is tested by loading the frame buffer (FB) memory with four neutral data patterns, then setting a target FB pixel with data that triggers the test mode readback latch. The result is read from the readback register after vertical blanking occurs. Two different trigger patterns are used at each FB pixel. All four MDI pixel paths (A - D) are used, and the pixel locations for each trigger are designed to detect gross MDI input data opens or short, VRAM SAM addressing, and VRAM-to-SAM transfer addressing. The screen shows four horizontal bars, which are either greyscale or colored. These bars change each time the trigger data is inverted, and as it completes the testing of a raster pattern.  
  
  Note: If the resolution and VRAM size permits, 8-bits per pixel mode are tested. |
Driver (Group 9)  Test all IOCTL calls that have not yet been used to verify proper driver communication to the hardware. Call the driver to perform a hardware update, and then confirm that the update was successful by using the complementary driver read, or reading the mmap’ed address space and comparing it against the stimulus.

cg14test performs the appropriate steps before and after each test (if possible) to maintain context and prevent visual confusion by saving the register data before it is overwritten, disabling video (if possible), performing the specific test, and restoring the saved register data information.

The data used for register testing is optimized to include all 0’s, all 1’s, and walking a 1 through each bit being tested.

<table>
<thead>
<tr>
<th>Test Groups</th>
<th>Description</th>
</tr>
</thead>
</table>
| MDI and VBC Chip Control Registers (Group 1) | • Master Control Register bits 7-0 write/read verify  
• Packed Pixel Register bits 3-0 write/read verify  
• Master Status Register bits 7-4 read-only verify 0x00 and 0x30 occur  
• Horizontal Blank Start Register bits 9-0 write/read verify  
• Horizontal Blank Clear Register bits 9-0 write/read verify  
• Horizontal Sync Set Register bits 9-0 write/read verify  
• Horizontal Sync Clear Register bits 9-0 write/read verify  
• Composite Sync Clear Register bits 9-0 write/read verify  
• Vertical Blank Start Register bits 11-0 write/read verify  
• Vertical Blank Clear Register bits 11-0 write/read verify  
• Vertical Sync Set Register bits 11-0 write/read verify  
• Vertical Sync Clear Register bits 11-0 write/read verify  
• Transfer Cycle Set Register bits 9-0 write/read verify (MDI revision 0 only)  
• Transfer Cycle Clear Register bits 9-0 write/read verify (MDI revision 0 only)  
• Fault Status Address Register bits 15-0 write/read verify  
• Auto-increment Address Space Register bits 7-0 write/read verify  
• Video Base Register bits 23-12 write/read verify |
Memory Chips

The Memory Chips test group includes VRAM Testing, Memory Retention, and Test Write Recovery.

VRAM Testing:
The Data Bus test uses 18 NTA patterns (Nair, Thatte, and Abraham’s testing procedure for RAM) to check for data and address faults. This test is performed in MDI_CHUNKY_XBGR_MAP access mode only. See TABLE 8-2.
The test ascends through the frame buffer memory, clearing it to 0’s. The NTA pattern test number $x$ reads a location to make sure test data $y$ is present. It then writes new data $z$ to that location. The location ascends through the FB sequentially. See TABLE 8-2.

Memory Retention:
The VRAM Data Retention test checks for gross problems with the VRAM refresh. Since refresh is active during this test, no retention problems should occur unless the refresh is defective.

This test turns off the video, writes 0’s to all the VRAM, waits the specified memory hold time (the default is five seconds), then reads and compares all VRAM data. This process is repeated with data of f’s, then the video is restored and the test is complete.

Two command-line parameters are related to this test: \texttt{R=number} and \texttt{H=number}. \texttt{R=} lets the user specify the refresh interval from 128-1023. The time between refresh cycles and the system default is 123. \texttt{H=} lets the user specify the retention test hold time in seconds.

Write Recovery:
A Write Recovery test is used in all the EMC mapping modes to write data to 0’s followed by immediately reading that data location to see if the VRAM can recover from a write correctly. This is done to all sequential ascending locations. Next, a second independent pass of memory is made with the complementary data of $0xffffffff$ written to descending locations of the FB memory buffer.

The EMC mapping access modes are:
- MDI_CHUNKY_XGBR_MAP
- MDI_CHUNKY_BGR_MAP
- MDI_PLANAR_X16_MAP
- MDI_PLANAR_C16_MAP
- MDI_PLANAR_X32_MAP
- MDI_PLANAR_B32_MAP
- MDI_PLANAR_G32_MAP
- MDI_PLANAR_R32_MAP

<table>
<thead>
<tr>
<th>Test Groups</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Chips (Group 2)</td>
<td>VRAM Testing (Continued)</td>
</tr>
<tr>
<td>Continued</td>
<td>The test ascends through the frame buffer memory, clearing it to 0’s. The NTA pattern test number $x$ reads a location to make sure test data $y$ is present. It then writes new data $z$ to that location. The location ascends through the FB sequentially. See TABLE 8-2.</td>
</tr>
</tbody>
</table>
### TABLE 8-1  cg14 Test Groups (Continued)

<table>
<thead>
<tr>
<th>Test Groups</th>
<th>Description</th>
</tr>
</thead>
</table>
| MDI Chip Cursor Registers (Group 3) | The MDI Chip Cursor Registers are:  
- Cursor Plane 0 Register bits 31-0 write/read verify  
- Cursor Plane 1 Register bits 31-0 write/read verify  
- Cursor Plane 0 Register bits 31-0 write/read verify (with auto increment)  
- Cursor Plane 1 Register bits 31-0 write/read verify (with auto increment)  
- Cursor Control Register bits 2-0 write/read verify  
- Cursor Color Register 1 bits 28-0 write/read verify  
- Cursor Color Register 2 bits 28-0 write/read verify  
- X-Cursor Location Register bits 11-0 write/read verify  
- Y-Cursor Location Register bits 11-0 write/read verify  
- Cursor Plane 0 Non-Auto Registers test  
- Cursor Plane 0 Auto Registers test  
- Cursor Plane 1 Non-Auto Registers test  
- Cursor Plane 1 Auto Registers test  
- Cursor Planes Retry A test  
- Cursor Planes Retry B test |
| MDI Chip CLUT Registers (Group 4) | The MDI Chip CLUT Registers are:  
- LUT1 Registers 0-255 bits 31-27 & 23-0 write/read verify  
- LUT1 Registers 0-255 bits 31-27 & 23-0 write/read verify (with auto increment)  
- LUT1D Registers 0-255 bits 31-27 & 23-0 write/read verify  
- LUT1D Registers 0-255 bits 31-27 & 23-0 write/read verify (with auto increment)  
- LUT2 Registers 0-255 bits 31-27 & 23-0 write/read verify  
- LUT2 Registers 0-255 bits 31-27 & 23-0 write/read verify (with auto increment)  
- LUT2D Registers 0-255 bits 31-27 & 23-0 write/read verify  
- LUT2D Registers 0-255 bits 31-27 & 23-0 write/read verify (with auto increment)  
- LUT3 Registers 0-255 bits 31-27 & 23-0 write/read verify  
- LUT3 Registers 0-255 bits 31-27 & 23-0 write/read verify (with auto increment)  
- LUT3D Registers 0-255 bits 31-27 & 23-0 write/read verify  
- LUT3D Registers 0-255 bits 31-27 & 23-0 write/read verify (with auto increment) |
The DAC Chip Registers test group includes the RAMDAC registers and control registers.

**RAMDAC Registers:**
- Address Register bits 7-0 (0x7 maximum) write/read verify
- Mode Register bits 7-0 (skip bit 5) bits write/read verify

**Control Registers:**
- ID Register bits 7-0 r/o verify data is 0x8C
- Pixel-Mask Register bits 7-0 write/read verify (skipped if dac rev = 2)
- Command2 Register bits 7-0 write/read verify (skipped if dac rev = 2)
- Command3 Register bits 7-0 write/read verify (skipped if dac rev = 2)

The MDI Chip XLUT Registers are:
- XLUT Registers 0-255 bits 7-0 write/read verify
- XLUT Registers 0-255 bits 7-0 write/read verify (with auto increment)
- XLUTD Registers 0-255 bits 7-0 write/read verify
- XLUTD Registers 0-255 bits 7-0 write/read verify (with auto increment)

<table>
<thead>
<tr>
<th>Test Groups</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAC Chip Registers (Group 5)</td>
<td>The DAC Chip Registers test group includes the RAMDAC registers and control registers.</td>
</tr>
<tr>
<td></td>
<td>RAMDAC Registers:</td>
</tr>
<tr>
<td></td>
<td>• Address Register bits 7-0 (0x7 maximum) write/read verify</td>
</tr>
<tr>
<td></td>
<td>• Mode Register bits 7-0 (skip bit 5) bits write/read verify</td>
</tr>
<tr>
<td></td>
<td>Control Registers:</td>
</tr>
<tr>
<td></td>
<td>• ID Register bits 7-0 r/o verify data is 0x8C</td>
</tr>
<tr>
<td></td>
<td>• Pixel-Mask Register bits 7-0 write/read verify (skipped if dac rev = 2)</td>
</tr>
<tr>
<td></td>
<td>• Command2 Register bits 7-0 write/read verify (skipped if dac rev = 2)</td>
</tr>
<tr>
<td></td>
<td>• Command3 Register bits 7-0 write/read verify (skipped if dac rev = 2)</td>
</tr>
<tr>
<td>MDI Chip XLUT Registers (Group 6)</td>
<td>The MDI Chip XLUT Registers are:</td>
</tr>
<tr>
<td></td>
<td>• XLUT Registers 0-255 bits 7-0 write/read verify</td>
</tr>
<tr>
<td></td>
<td>• XLUT Registers 0-255 bits 7-0 write/read verify (with auto increment)</td>
</tr>
<tr>
<td></td>
<td>• XLUTD Registers 0-255 bits 7-0 write/read verify</td>
</tr>
<tr>
<td></td>
<td>• XLUTD Registers 0-255 bits 7-0 write/read verify (with auto increment)</td>
</tr>
</tbody>
</table>
### TABLE 8-1  cg14 Test Groups (Continued)

<table>
<thead>
<tr>
<th>Test Groups</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG14 Display (visual only) Group 7)</td>
<td>This test visually displays 256 boxes on the screen (each in a different color), and then shifts the CLUT1 entries giving the visual impression of the pattern mirroring itself from left to right horizontally. The pattern then rotates up, down, followed by mirroring itself horizontally left to right.</td>
</tr>
<tr>
<td>MDI Chip Test Mode Readback Register (Group 8)</td>
<td>This test mode reads back register bits 23-0 in read-only and verify modes.</td>
</tr>
</tbody>
</table>
Driver IOCTLs (Group 9)

- MDI_GET_CFGINFO check # of CLUT’s, pixel height, pixel width, and pixel mode against hardware
- FBIOGATTR check real_type, fb_height, fb_width, fb_depth, fb_cmsize, and fb_size against cfginfo values
- FBIOGTYPE check fb_type, fb_height, fb_width, fb_depth, fb_size, and fb_cmsize against driver defines or cfginfo values
- FBIOGVIDEO check status returned against hardware
- FBIOSSVIDEO set off, off, on, on, off verifying against hardware
- FBIOVERTICAL (imbedded in FBIOSSVIDEO)
- MDI_VRT_CNTL turn off, on, on, off the video interrupt enable and verify the hardware agrees
- MDI_SET_PIXELMODE set different modes and verify against the hardware
- MDI_SET_PPR set the different modes and verify against the hardware
- MDI_SET_COUNTERS set HSS, HSC, XCC, HBC, XCS, HBS, CSC, VSS, VSC, VBC, VBS, HCT, and VCT then verify against hardware
- MDI_SET_XLUT set xlut and verify against hardware
- MDI_GET_XLUT get xlut and verify against hardware
- MDI_SET_CLUT set clut (1-3 as applicable) and verify against hardware
- MDI_GET_CLUT get clut (1-3 as applicable) and verify against hardware
- FBIOPUTCMAP set and verify clut1 matches
- FBIOGETCMAP verify clut1 matches get
- FBIOATTR set emu_type to FBTYME_MDICOLOR and verify
- FBIOATTR check
- FBIOGCMRMAX verify x and y size match driver defines
- FBIOSCURSOR verify set at 3 locations matches hardware
- FBIOGSCURSOR verify driver knows what set(s) just did
- FBIOSCURPOS verify set at three locations matches hardware
- FBIOGSCURPOS verify driver knows what set(s) just did
- MDI_SET_CURSOR set then check CCR, XCU, and YCU cursor hardware registers

TABLE 8-1  cg14 Test Groups (Continued)

<table>
<thead>
<tr>
<th>Test Groups</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver IOCTLs (Group 9)</td>
<td></td>
</tr>
</tbody>
</table>

- MDI_GET_CFGINFO check # of CLUT’s, pixel height, pixel width, and pixel mode against hardware
- FBIOGATTR check real_type, fb_height, fb_width, fb_depth, fb_cmsize, and fb_size against cfginfo values
- FBIOGTYPE check fb_type, fb_height, fb_width, fb_depth, fb_size, and fb_cmsize against driver defines or cfginfo values
- FBIOGVIDEO check status returned against hardware
- FBIOSSVIDEO set off, off, on, on, off verifying against hardware
- FBIOVERTICAL (imbedded in FBIOSSVIDEO)
- MDI_VRT_CNTL turn off, on, on, off the video interrupt enable and verify the hardware agrees
- MDI_SET_PIXELMODE set different modes and verify against the hardware
- MDI_SET_PPR set the different modes and verify against the hardware
- MDI_SET_COUNTERS set HSS, HSC, XCC, HBC, XCS, HBS, CSC, VSS, VSC, VBC, VBS, HCT, and VCT then verify against hardware
- MDI_SET_XLUT set xlut and verify against hardware
- MDI_GET_XLUT get xlut and verify against hardware
- MDI_SET_CLUT set clut (1-3 as applicable) and verify against hardware
- MDI_GET_CLUT get clut (1-3 as applicable) and verify against hardware
- FBIOPUTCMAP set and verify clut1 matches
- FBIOGETCMAP verify clut1 matches get
- FBIOATTR set emu_type to FBTYME_MDICOLOR and verify
- FBIOATTR check
- FBIOGCMRMAX verify x and y size match driver defines
- FBIOSCURSOR verify set at 3 locations matches hardware
- FBIOGSCURSOR verify driver knows what set(s) just did
- FBIOSCURPOS verify set at three locations matches hardware
- FBIOGSCURPOS verify driver knows what set(s) just did
- MDI_SET_CURSOR set then check CCR, XCU, and YCU cursor hardware registers
### TABLE 8-2  cg14test NTA Testing Patterns

<table>
<thead>
<tr>
<th>NTA Test Pattern Number = x</th>
<th>Test Data = y</th>
<th>New Data = z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0x00000000</td>
<td>0x01010101</td>
</tr>
<tr>
<td>1.5</td>
<td>0x01010101</td>
<td>0xffffffff</td>
</tr>
<tr>
<td>2.1</td>
<td>0xffffffff</td>
<td>0x1f1f1f1f</td>
</tr>
<tr>
<td>2.2</td>
<td>0x33333333</td>
<td>0x00000000</td>
</tr>
<tr>
<td>3.1</td>
<td>0x00000000</td>
<td>0x01010101</td>
</tr>
<tr>
<td>3.2</td>
<td>0x0f0f0f0f</td>
<td>0x00000000</td>
</tr>
<tr>
<td>4.1</td>
<td>0x0f0f0f0f</td>
<td>0x55555555</td>
</tr>
<tr>
<td>4.2</td>
<td>0x55555555</td>
<td>0xaaaaaaa</td>
</tr>
<tr>
<td>5.1</td>
<td>0xaaaaaaa</td>
<td>0x05050505 (1x)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x88888888 (2x)</td>
</tr>
<tr>
<td>5.2</td>
<td>0x88888888</td>
<td>0xf5f5f5f5</td>
</tr>
<tr>
<td>6.1</td>
<td>0xf5f5f5f5</td>
<td>0x00000000 (1x)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x5f5f5f5f (2x)</td>
</tr>
<tr>
<td>6.2</td>
<td>0x5f5f5f5f</td>
<td>0x11111111</td>
</tr>
<tr>
<td>7.1</td>
<td>0x11111111</td>
<td>0x00000000 (1x)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xccccccccc (2x)</td>
</tr>
<tr>
<td>7.2</td>
<td>0xccccccccc</td>
<td>0xdbdbdbdb</td>
</tr>
<tr>
<td>8.1</td>
<td>0xdbdbdbdb</td>
<td>0x6d6d6d6d</td>
</tr>
<tr>
<td>8.2</td>
<td>0x6d6d6d6d</td>
<td>0x6b6b6b6b</td>
</tr>
<tr>
<td>9.1</td>
<td>0x6b6b6b6b</td>
<td>0x0000000</td>
</tr>
<tr>
<td>9.2</td>
<td>0x00000000</td>
<td>-</td>
</tr>
</tbody>
</table>

**cg14test Options**

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User’s Guide* for more details.
FIGURE 8-1  cg14test Test Parameter Options Dialog Box

TABLE 8-3  cg14test Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB Locking</td>
<td>See “Testing Frame Buffers” on page 18 for details.</td>
</tr>
<tr>
<td>Long Test</td>
<td>When enabled, the color bar screen(s) in the MDI Testmode Readback test checks all SAM transfers in clock=0 mode and clock=1 mode. If Long test is disabled, clock=1 runs checks on the first eight addresses and the first SAM transfer only.</td>
</tr>
<tr>
<td>Processor Affinity</td>
<td>For multiprocessor systems, indicates the processor to be tested.</td>
</tr>
</tbody>
</table>
cg14test Test Modes

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

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>This mode uses all subtests to test the cg14 frame buffer. The user can select the long mode for TRMB subtest.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

cg14test Command-Line Syntax

`/opt/SUNWvts/bin/cg14test standard_arguments -o dev=device_name, lock= E(nable)/D(isable),L,I`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the path of the cg14 device file to be tested; for example: <code>/dev/fbs/device_name</code>.</td>
</tr>
<tr>
<td>lock= E(nable)/D(isable)</td>
<td>Enables and disables the window system locking option. See “Testing Frame Buffers” on page 18 for details. Do not use when your device is the window system display.</td>
</tr>
<tr>
<td>L</td>
<td>Enables the long TMRB test.</td>
</tr>
<tr>
<td>I</td>
<td>Enables optional driver ioctl tests for cursor. Note: Do not move the mouse during the cg14test when you run this option.</td>
</tr>
</tbody>
</table>

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 “32-Bit and 64-Bit Tests” on page 13.
Frame Buffer, GX, GXplus, and TurboGX Options Test (cg6)

The cg6 test verifies the GX, GXplus, and TurboGX™ frame buffer and the graphics options offered with most SPARC™ based workstations and servers. This test stresses the frame buffer with the subtests described below.

For full instructions on testing frame buffers, see “Testing Frame Buffers” on page 18.

**Note** – Disable all screen savers before testing any graphics device. Type `xset s off` at the prompt to disable the Solaris screen saver.

To start SunVTS with vtsui, but without vtsk, add the host name to `xhost` as: `xhost + hostname`. 
## cg6 Subtests

<table>
<thead>
<tr>
<th>Subtests</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cursor test</strong></td>
<td>cg6 visually checks the overlay registers of the RAMDAC. A pointer is drawn on the screen and moved to predetermined locations. There is a problem if the pointer disappears. This visual test ensures that the overlay is working properly.</td>
</tr>
<tr>
<td><strong>Fast Copy in double buffer test mode</strong></td>
<td>Creates two full-size screen raster images in double buffer mode, writing different patterns to each. The hidden buffer is copied to the visible buffer, and the data is compared. An error message is returned if there are inconsistencies. Then the buffer is flipped and the process is repeated.</td>
</tr>
<tr>
<td><strong>TEC test</strong></td>
<td>Verifies that the transformation engine and cursor control logic are accessible to confirm that further TEC access is performed correctly.</td>
</tr>
<tr>
<td><strong>FBC test</strong></td>
<td>Verifies that the frame buffer controller logic is accessible to confirm that further FBC access is performed correctly.</td>
</tr>
<tr>
<td><strong>Frame Buffer test</strong></td>
<td>Verifies that the frame buffer memory is working. A “walking ones” pattern is written to memory, with a specific color signifying one of eight bits. The screen is divided into eight equally wide vertical stripes. A “walking one” is written to each stripe, causing eight iterations of these stripes. The value written is read back and checked. If the values do not match, an error is reported.</td>
</tr>
<tr>
<td><strong>Screen test using blits</strong></td>
<td>Draws blocks of color and performs blit transfers to other portions of the screen. First, the entire screen is drawn with cyan, then a black block is placed in the upper-left corner. This subtest blits this block on the upper-right, lower-right, and lower-left corners, then “or’s” the whole image.</td>
</tr>
<tr>
<td><strong>Blit test</strong></td>
<td>Draws a block of data and blit into a location at the bottom-right rectangle.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Subtests</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line test</td>
<td>Draws lines on the screen in different colors with different data values. The data is read back and compared with the expected values. An error is returned in the case of a mismatch.</td>
</tr>
<tr>
<td>Polygon test</td>
<td>Draws hourglass-shaped polygons on the screen, using the four vertices. After all the polygons are rendered in the video memory, they are read back and the data compared with expected values. If there is a mismatch, an error is displayed.</td>
</tr>
<tr>
<td>Colormap test</td>
<td>Loads all 256 locations in the color map with a greyscale, both backward and forward. This means decreasing values are loaded to all R, G, and B values.</td>
</tr>
</tbody>
</table>

Note: If the system being tested has a monochrome or greyscale monitor, visual color problems are undetectable.
Note – See “Testing Frame Buffers” on page 18 for details about frame buffer locking.

**FIGURE 9-1**  cg6 Test Parameter Options Dialog Box

### cg6 Test Modes

Due to the nature of graphic tests, reading from or writing to the frame buffer during graphic tests will disturb user operation.

**TABLE 9-2**  cg6 Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests</td>
</tr>
</tbody>
</table>
cg6 Command-Line Syntax

/opt/SUNWvts/bin/cg6 standard_arguments -o dev=device_name, lock=E(nable)/D(isable), Passes=number

TABLE 9-3  cg6 Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the path of the cg14 device file to be tested,</td>
</tr>
<tr>
<td></td>
<td>for example, /dev/fbs/device_name.</td>
</tr>
<tr>
<td>lock=E(nable)/D(isable)</td>
<td>Enables/disables the window system locking option. See</td>
</tr>
<tr>
<td></td>
<td>“Testing Frame Buffers” on page 18 for details. Do not use</td>
</tr>
<tr>
<td></td>
<td>when device is the window system display.</td>
</tr>
<tr>
<td>Passes=number</td>
<td>The number of passes to run. The default is 1.</td>
</tr>
</tbody>
</table>

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 “32-Bit and 64-Bit Tests” on page 13.

Note – Extra swap space of 5 MB is required.
CPU Test (cputest)

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

The cputest comprises two subtests:

- g0 subtest—tests a processor’s g0 register functionality. The g0 subtest is only supported on UltraSPARC-based systems.

- CUC subtest—tests a processor’s ability to correctly execute a Compress/Uncompress/Compare (cmp) command sequence on machines with the SPARC-V9 architecture.

As the CUC subtest runs, it creates four files in the /tmp/sunvts directory. The CUC subtest uses the following file naming conventions where nnn represents the processor unit number and x represents a random character string appended to the file name:

- PnnnPx—original pattern file (size determined by the cputest File Size option)
- PnnnZx—compressed version of the pattern file
- PnnnUX—uncompressed data from the PnnnZx file
- PnnnCX—comparison data between the PnnnPx and PnnnUX files

With the cputest file retention mode, you can control whether these files are deleted or not, so that in the event of a miscompare, you can view the contents of the files to analyze the miscompared data. The exact names of the files are displayed in the SunVTS message window whenever the files are saved such as when there is a compression miscompare or when the file retention mode is set to save. Refer to “cputest Options” on page 1074 for more details.

**Caution**—Do not run the CUC subtest with the retention mode set to save for numerous passes, otherwise the files that are saved in /tmp/sunvts can fill the /tmp capacity. If /tmp is mounted to the swap area, the swap space may become filled to capacity.
Note – Only one instance of cputest per processor is possible.

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

Note – The cputest only runs on SPARC V9 systems.

cputest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 10-1  cputest Test Parameter Options Dialog Box
### TABLE 10-1  
**cputest** Option Dialog Box Descriptions

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>Specifies the number of times to loop on the selected subtests. Use the up/down arrows to select a value from 1 to 8192. The default varies depending on the SunVTS test mode.</td>
</tr>
</tbody>
</table>
| Test List            | Specifies which subtests to run. The choices are:  
  • CUC—the compress/uncompress/compare subtest.  
  • g0—the g0 register subtest.  
  Refer to the general test description at the beginning of this chapter for subtest descriptions. If no subtest is selected, both subtests run. |
| File Size (KBytes)   | Specifies the size of the CUC pattern file in KBytes. Select a value from 1 to 8192. The default varies depending on the SunVTS test mode. |
| File Retention Mode  | Specifies whether the **cputest** removes the CUC pattern files or not. The choices are:  
  • Purge—unconditionally removes the four subtest files.  
  • Save—does not remove any of the four subtest files.  
  • OnError—removes the four subtest files unless the CUC resulted in a miscompare. In this case, do not remove the files.  
  The default is OnError.  
  Refer to the Caution at the beginning of this chapter regarding the Save value. |
| Processor Affinity   | Although the test parameter dialog box displays the processor affinity “bound to” selection box, the processor that corresponds to this instance of the **cputest** is determined when the SunVTS kernel probes for devices. Therefore, switching processor affinity in this dialog box is not supported. |
cputest Test Modes

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

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Both subtests are selected. The test options are fixed with the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Iterations=5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• File Size=64 KBytes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• File retention=OnError</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Both subtests are selectable, and all the test options are available to scale the cputest as needed.</td>
</tr>
</tbody>
</table>

cputest Command-Line Syntax

```
/opt/SUNWvts/bin/cputest standard_arguments -o dev=device_name,count=count_number,test=testlist,size=file_size,retain=mode
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the name of the device to test, for example, cpu-unit5</td>
</tr>
<tr>
<td>count=count_number</td>
<td>Defines the number of times to loop on the subtests. Use a number from 1 to 8192. The default is 200.</td>
</tr>
</tbody>
</table>
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 “32-Bit and 64-Bit Tests” on page 13.
CHAPTER 11

CPU Power Management Test
(cpupmtest)

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

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

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

copupmtest is currently supported on Sun Blade™ 100 and Sun Blade 1000 systems.

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

TABLE 11-1 cpupmtest Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level1</td>
<td>Sets the number of minutes to test the system at its lowest speed.</td>
</tr>
<tr>
<td>Level2</td>
<td>Sets the number of minutes to test the system at a level higher than the lowest speed.</td>
</tr>
<tr>
<td>Level3</td>
<td>Sets the number of minutes to test the system at a level higher than Level2 speed.</td>
</tr>
</tbody>
</table>
TABLE 11-1  cpupmtest Options (Continued)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level&lt;n-1&gt;</td>
<td>Sets the number of minutes to test the system at a level lower than the normal speed.</td>
</tr>
<tr>
<td>Level&lt;n&gt;</td>
<td>Sets the number of minutes to test the system at its normal speed.</td>
</tr>
<tr>
<td>Log Power States</td>
<td>Records and timestamps every CPU power level change to the VTS log file (/var/opt/SUNWvts/logs/sunvts.info).</td>
</tr>
</tbody>
</table>

---

**cpupmtest Test Modes**

TABLE 11-2  cpupmtest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full test.</td>
</tr>
</tbody>
</table>

---

**cpupmtest Command-Line Syntax**

```
/opt/SUNWvts/bin/cpupmtest  standard_arguments
-o  dev=cpupm, speed1=mn, speed2=mn, speed3=mn
```

TABLE 11-3  cpupmtest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=cpupm</td>
<td>Specifies the name of the device.</td>
</tr>
<tr>
<td>speed1=mn</td>
<td>Sets the number of minutes to test the system at its lowest speed.</td>
</tr>
<tr>
<td>speed2=mn</td>
<td>Sets the number of minutes to test the system at the next level of speed.</td>
</tr>
<tr>
<td>speed3=mn</td>
<td>Sets the number of minutes to test the system at its highest speed.</td>
</tr>
</tbody>
</table>
Disk and Floppy Drives Test
(disktest)

disktest verifies the functionality of hard drives and diskette drives using three subtests (see TABLE 12-1): Media, File System, and Asynchronous I/O. Most disk drives, such as SCSI disks, native or SCSI floppy disks, IPI, and so on, are supported. The type of drive being tested is displayed at the top of the Test Parameter Options dialog box.

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

disktest Test Requirements

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

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

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

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

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

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

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

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

To use option files with disktest, create two separate option files for the two different states of the BYPASS_FS_PROBE environment variable.
TABLE 12-1 describes the disktest subtests.

**TABLE 12-1 disktest Subtests**

<table>
<thead>
<tr>
<th>Subtest</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media subtest</td>
<td>Verifies disk media by writing data to and reading data from the disk. The Media subtest treats a disk as one large chunk of contiguous data.</td>
</tr>
<tr>
<td></td>
<td>This is a scalable test that can run multiple copies in read/write mode on the same disk partition. To avoid data corruption, all simultaneous instances of disktest communicate through a shared memory service. This ensures that different copies of the media subtest do not overlay the same disk block at the same time.</td>
</tr>
<tr>
<td></td>
<td>The Media subtest runs in two different modes as described below:</td>
</tr>
<tr>
<td></td>
<td><strong>SyncIO:</strong> The SyncIO media test creates a random offset for a partition from which the media testing starts. Starting from this offset, the read (Readonly mode) or the write/read (WriteRead mode) begins in a sequential fashion. The test continues until the specified percentage of media is covered.</td>
</tr>
<tr>
<td></td>
<td><strong>AsyncIO:</strong> The AsyncIO media test always starts from the first block of the partition under test and covers the area specified by the Media Coverage percentage.</td>
</tr>
<tr>
<td></td>
<td>AsyncIO uses the read/write feature of the Solaris disk driver to exercise the disk. In Readonly mode, the test sends a maximum of four asynchronous read packets, each with a random size and a random offset into the selected partition. 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 is tested. In WriteRead mode, one write packet is issued in every four read packets as a spot check of the write operation. Before data is written to a particular location, data is backed up, write-verified, and restored to its original state.</td>
</tr>
<tr>
<td>File System subtest</td>
<td>Verifies the disk system’s integrity. The File System subtest exercises the partition being tested to determine if it is mounted. If the partition is not already mounted or pre-mounted, then the test is blocked. The test opens two temporary files (of the size specified on File System File Size) and performs a Read/Write test.</td>
</tr>
</tbody>
</table>
disktest Test Options

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

FIGURE 12-1 disktest Test Parameter Options Dialog Box
TABLE 12-2 describes the disktest option menu for different test modes.

**TABLE 12-2**  
disktest Configurations and Options

<table>
<thead>
<tr>
<th>disktest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>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.</td>
</tr>
<tr>
<td>Test Media</td>
<td>Enables or disables the Media subtest.</td>
</tr>
<tr>
<td>Media Write Read Mode</td>
<td>Enables Read-Only or Compare after Read or Read after Write, with or without backup.</td>
</tr>
<tr>
<td>Media Test Method</td>
<td>Enables or disables the Media Test Methods (SyncIO and AsyncIO).</td>
</tr>
<tr>
<td>Media Coverage (%)</td>
<td>Enables users to test all or part of a partition (in percentages).</td>
</tr>
<tr>
<td>Media Transfer Size</td>
<td>Displays the transfer size of the media subtest.</td>
</tr>
<tr>
<td>Test File System</td>
<td>Enables or disables the File System subtest.</td>
</tr>
<tr>
<td>File System File Size</td>
<td>Creates a file system file size twice the size of what is specified.</td>
</tr>
<tr>
<td>File System Transfer Size</td>
<td>Displays the transfer size of the File System subtest.</td>
</tr>
<tr>
<td>File System Test Pattern</td>
<td>Test pattern of File System subtest.</td>
</tr>
</tbody>
</table>
| Connection Test for Hard Disk | • Option Menu for hard disk partition—0 - 7 [default]  
• Test Media—[Enable](fixed to Enable)  
• Media Write Read Mode—[Read Only](fixed to Read Only)  
• Media Test Method—[SyncIO] (fixed to SyncIO)  
• Media Coverage(%)—1  
• Media Transfer Size—[2KB]  
• Test File System—[Disable](fixed to Disable) |
<table>
<thead>
<tr>
<th>disktest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Online Mode for Hard Disk</strong></td>
<td></td>
</tr>
<tr>
<td>Partition—0 - 7 [default]</td>
<td></td>
</tr>
<tr>
<td>Test Media—[Enable] [Disable]</td>
<td></td>
</tr>
<tr>
<td>Test Mode—[Read-only] [fixed to Read-only]</td>
<td></td>
</tr>
<tr>
<td>Media Coverage(%)—[10]</td>
<td></td>
</tr>
<tr>
<td>Media Transfer Size—[2KB]</td>
<td></td>
</tr>
<tr>
<td>Test File System—[Disable] [fixed to Disable]</td>
<td></td>
</tr>
<tr>
<td><strong>Functional Test for Hard Disk</strong></td>
<td></td>
</tr>
<tr>
<td>Partition—0 - 7 [default]</td>
<td></td>
</tr>
<tr>
<td>Test Media—[Enable] [Disable]</td>
<td></td>
</tr>
<tr>
<td>Media Write Read Mode—[Readonly] [CompareRead] [WriteRead]</td>
<td></td>
</tr>
<tr>
<td>Media Test method—[SyncI0] [AsyncI0]</td>
<td></td>
</tr>
<tr>
<td>Media Coverage(%)—[30]</td>
<td></td>
</tr>
<tr>
<td>Media Transfer Size—[2KB] [16KB] [32KB] [64KB] [128KB] [256KB] [512KB]</td>
<td></td>
</tr>
<tr>
<td>Test File System—[Enable] [Disable]</td>
<td></td>
</tr>
<tr>
<td>File System File Size—[512KB] [2MB] [8MB] [20MB] [100MB] [200MB]</td>
<td></td>
</tr>
<tr>
<td>File System Transfer Size—[512B] [1024B] [10KB] [40KB] [80KB]</td>
<td></td>
</tr>
<tr>
<td>File System Test Pattern—[sequential] [0x00000000] [0xffffffff] [0x5aa55aa5] [0xdb6db6db] [random]</td>
<td></td>
</tr>
<tr>
<td><strong>Functional Test for Floppy Disk</strong></td>
<td></td>
</tr>
<tr>
<td>(under Other-Devices group)—partition: 0 - 7 [default]</td>
<td></td>
</tr>
<tr>
<td>Test Media—[Enable] [Disable]</td>
<td></td>
</tr>
<tr>
<td>Media Write Read Mode—[Read-only] [BackupWriteRead]</td>
<td></td>
</tr>
<tr>
<td>Media Test Method—[SyncI0] [AsyncI0]</td>
<td></td>
</tr>
<tr>
<td>Media Coverage(%)—[30]</td>
<td></td>
</tr>
<tr>
<td>Media Transfer Size—[2KB] [10KB] [20KB]</td>
<td></td>
</tr>
<tr>
<td>Test File System—[Enable] [Disable]</td>
<td></td>
</tr>
<tr>
<td>Floppy File Size— [100KB] [200KB]</td>
<td></td>
</tr>
<tr>
<td>Floppy Transfer Size—[512B] [1024B] [10KB]</td>
<td></td>
</tr>
<tr>
<td>File System Test Pattern—[sequential] [0x00000000] [0xffffffff] [0x5aa55aa5] [0xdb6db6db] [random]</td>
<td></td>
</tr>
</tbody>
</table>
disktest Test Modes

TABLE 12-3  

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>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 offline Functional test mode.</td>
</tr>
</tbody>
</table>

disktest Command-Line Syntax

/opt/SUNWvts/bin/disktest  
standard_arguments  
-o  
dev=dev=device_name  
,partition=<0-7>"(mount_point)" ,rawsub=E | D , 
rawrw=Readonly | CompareRead | WriteRead ,  
method=AsyncIO+SyncIO , rawcover=n ,  
rawiosize=n , fssub=E | D , fssize=n , fsiosize=n , fspattern=data_pattern

TABLE 12-4  

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the name of the disk to be tested, such as c0t3d0.</td>
</tr>
</tbody>
</table>
| partition=n"(mount_point)" | Specifies the partition number as follows:  
•  n—is the partition number (slice number), usually 0-7  
•  mount_point—is the mount point for the mounted partition that you plan to test  
For example: partition=6"(/export)"
| rawsub=enable | Enables or disables the media subtest. |
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 “32-Bit and 64-Bit Tests” on page 13.
Sun Fire 880 FC-AL Disk Backplane Test (dpmtest)

dpmtest tests the Sun Fire™ 880 FC-AL disk backplane. dpmtest verifies the operation of the embedded SES controllers and the disk enclosure system of the Sun Fire 880 workgroup server.

No special hardware is required to run the dpmtest test.

dpmtest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 13-1  dpmtest Test Parameter Options Dialog Box
**TABLE 13-1**  
dpctest Test Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Path</td>
<td>Default: Test runs using the interface through which the test was registered. If the device tested is an SES device, then the dpctest tests the device over the fiber channel interface. If the device is registered as an SSC100 device, then it tests over the I2C interface. Both: If selected, the test runs the same tests to this device over both the Fiber channel interface and the I2C interface. Note: This option will be grayed out if both paths are not available.</td>
</tr>
<tr>
<td>Loopback Test</td>
<td>When enabled, the subtest will cause the SES device to loop a packet around the fiber bus with varying data patterns. The device reads the packet after the packet is received, and verifies that the data payload is correct. Note: This test grays out if the SSC100 does not have access to the fiber bus.</td>
</tr>
<tr>
<td>Firmware Test</td>
<td>When enabled, this subtest runs the system friendly embedded firmware tests on the SES/SSC100 device.</td>
</tr>
</tbody>
</table>

**TABLE 13-2**  
dpctest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>The test opens the device, extracts information about the device (firmware revision, drives installed, temperatures, etc) and displays the information for the user. Once the test is run, the device closes and the test exits.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>The test opens the device and runs the selected tests against the device. When fully run, the test closes and reports the results.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Chapter 13  Sun Fire 880 FC-AL Disk Backplane Test (dpctest)  93
**dpmtest Command-Line Syntax**

```
/opt/SUNWvts/bin/dpmtest standard_arguments -0 dev=[device name],
path=[Default|Both], lb=[Enable|Disable], fwtest=[Enable|Disable]
```

**TABLE 13-3 dpmtest Command-Line Syntax**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-0 dev</code></td>
<td><code>[device name]</code> is the path name of the device being tested.</td>
</tr>
<tr>
<td><code>path</code></td>
<td>Default: Test runs using the interface through which the test was registered. If the device tested is an SES device, then the <code>dpmtest</code> tests the device over fiber channel interface. If the device is registered as an SSC100 device, then it tests over the I2C interface. Both: If selected, the test runs the same tests to this device over both the Fiber channel interface and the I2C interface. Note: This option will be grayed out if both paths are not available.</td>
</tr>
<tr>
<td><code>lb</code></td>
<td>When enabled, the subtest will cause the SES device to loop a packet around the fiber bus with varying data patterns. The device reads the packet after the packet is received, and verifies that the data payload is correct. Note: This test will be grayed out in the case where the SSC100 does not have access to the fiber bus.</td>
</tr>
<tr>
<td><code>fwtest</code></td>
<td>When enabled, this subtest runs the system friendly embedded firmware tests on the SES/SSC100 device.</td>
</tr>
</tbody>
</table>
DVD Test (dvdtest)

The dvdtest tests the DVD by reading a DVD-ROM in the drive.

The dvdtest runs even if the Volume Manager is not running. If the Volume Manager is running and no media (DVD or CD) is installed in the drive, SunVTS prompts you to install media in the drive before selecting the test.

dvdtest Test Requirements

**Note** – Load a DVD-ROM into the DVD drive before running this test or the test will fail.

When a DVD-ROM is loaded in the drive, SunVTS uses the dvdtest to test the drive. When a CD (non-DVD type) is loaded, SunVTS uses the cdtest to test the drive. 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 will associate the correct test (dvdtest or cdtest) based on the media that is loaded in the drive.

dvdtest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User’s Guide* for more details.
FIGURE 14-1  dvdtest Test Parameter Options Dialog Box

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Data/Track</td>
<td>Specifies a percentage of data to test for each track. Use a number between 0 and 100.</td>
</tr>
<tr>
<td>Read Mode</td>
<td>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.</td>
</tr>
</tbody>
</table>
dvdtest Test Modes

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

**TABLE 14-2** dvdtest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>

---

**dvdtest Command-Line Syntax**

```bash
/opt/SUNWvts/bin/dvdtest standard_arguments
-o dev=device_name,read=random|sequential,data=%_of_data
```

**TABLE 14-3** dvdtest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dev=device_name</code></td>
<td>Specifies the name of the device to test, for example /dev/rdsk/cntn0.</td>
</tr>
<tr>
<td>`read=random</td>
<td>sequential`</td>
</tr>
<tr>
<td><code>data=%_of_data</code></td>
<td>Sets the percentage of data to test. Use a number from 0 to 100.</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
ECP 1284 Parallel Port Printer Test (ecpptest)

The `ecpptest` verifies the functionality of the `ecpp(7)` IEEE 1284 parallel printer port device.

The `ecpp(7)` device is an exclusive use device. Only one application can interface with it at a time.

This test is not scalable.

ecpptest Hardware and Software Requirements

The `ecpp(7)` driver is installed with the operating system if the system being tested supports the `ecpp(7)` device. To run the optional printer subtest, a Centronics or ECP-mode printer must be attached to the printer port. To run the optional external loopback test, a passive loopback connector must be installed on the printer port.

Note – The external loopback test is intended for Sun internal manufacturing use only. It requires a custom loopback connector not available to an external customer.
ecpptest Subtests

**TABLE 15-1  ecpptest Subtests**

<table>
<thead>
<tr>
<th>Subtest</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Test FIFO Loopback</td>
<td>Verifies DMA and PIO accesses to the ecpp device. It uses an Internal Test FIFO on the ecpp device. There are no printer or loopback connectors required. This test is always executed.</td>
</tr>
<tr>
<td>External Passive Loopback</td>
<td>This verifies the parallel port I/O connections to the back panel connector. This test requires a passive loopback connector (Sun part no. 270-2965-01). This test is disabled by default and must be manually enabled by the user. Note: The external loopback test is intended for Sun internal manufacturing use only. It requires a custom loopback connector that is not available to an external customer.</td>
</tr>
<tr>
<td>Printer test</td>
<td>This verifies the parallel port printer operation. It outputs a half page of ASCII character data. The output mode (for example, ECP and Centronics) is determined by which mode the printer and ecpp driver automatically negotiate. It will not change the current mode. The user must verify that data printed properly. This test is disabled by default and must be manually enabled by the user.</td>
</tr>
</tbody>
</table>

ecpptest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User’s Guide* for more details.
FIGURE 15-1  ecpptest Test Parameter Options Dialog Box

TABLE 15-2  ecpptest Options

<table>
<thead>
<tr>
<th>ecpptest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Loopback</td>
<td>Enables or disables the external loopback test. This test requires a special loopback plug.</td>
</tr>
<tr>
<td>Printer</td>
<td>Enables or disables the printer test; this test requires a printer to be attached to the parallel port.</td>
</tr>
</tbody>
</table>
### TABLE 15-2  ecpptest Options

<table>
<thead>
<tr>
<th>ecpptest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Printer_data_type</td>
<td>Choose whether ascii text or PostScript data is sent to the printer. The printer test must be enabled for this to be meaningful. A postscript printer must be attached to print postscript data.</td>
</tr>
<tr>
<td>Printer_delay _seconds</td>
<td>Allows the user to choose a delay between passes of the printer test. This prevents continuous printing of data that could quickly empty the paper supply. This is only meaningful if the printer test is enabled.</td>
</tr>
<tr>
<td>Internnal_loop_ during_delay</td>
<td>Enables the Internal Test FIFO loopback test during the printer delay time. This is only meaningful if the printer test is enabled.</td>
</tr>
</tbody>
</table>
ecpptest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Opens and closes the ecpp(7) device. No data is transferred. The test passes if the device can be open and closed successfully. The test also passes if the device cannot be opened because the device is busy with another process.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the internal loopback test and the user can optionally run the External loopback test and Printer test. The test will fail if the device is busy.</td>
</tr>
</tbody>
</table>

ecpptest Command-Line Syntax

```
/opt/SUNWvts/bin/ecpptest standard_arguments -o [dev=device_name] [ext_loop=Enable|Disable] [printer=Enable|Disable] [data=ascii|postscript] [delay=0-86400] [dloop=Enable|Disable]
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the name of the device. This should be of the form /dev/ecpp#, where # is the minor number of the device. The default device is /dev/ecpp0.</td>
</tr>
<tr>
<td>ext_loop=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>printer=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>data=ascii</td>
<td>postscript</td>
</tr>
<tr>
<td>delay=0-86400</td>
<td>Allows the user to choose a delay between passes of the printer test. This prevents continuous printing of data that could quickly empty the paper supply.</td>
</tr>
<tr>
<td>dloop=Enable</td>
<td>Disable</td>
</tr>
</tbody>
</table>
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 “32-Bit and 64-Bit Tests” on page 13.
 CHAPTER 16

Sun StorEdge A5x00 Test (enatest)

enatest is used to provide configuration verification, fault isolation, and repair validation of the Sun StorEdge™ A5x00 subsystem. The enatest tests Sun StorEdge models A5000 (14 slot disk array) and A5200 (22 slot disk array).

The Sun StorEdge A5x00 is a high availability mass storage subsystem consisting of:

- SCSI Fibre Channel protocol host adapters with dual 100-Megabyte FC-AL ports.
- A disk enclosure.
- A front panel display for configuration information.
- Up to two interface boards in the enclosure, which provide FC-AL connections to the enclosure and also provide status information and control of the conditions within the enclosure.
- Other field-replaceable units (FRUs) within the enclosure include power supply units, fan trays, and backplane.

**Note** – Do not run enatest and socaltest at the same time, otherwise test failures might occur.

**Note** – The Sun StorEdge A5x00 was formally known as the Sun Enterprise Network Array™. The enatest tests both of these disk array subsystems.
enatest detects all Sun StorEdge A5x00 enclosures connected to the host and collects relevant configuration information. FIGURE 16-1 shows the Test Parameter Options menu which contains a sample configuration listing and test parameters. TABLE 16-1 describes the extent of the test coverage and provides samples of the
configuration information that is displayed.

<table>
<thead>
<tr>
<th>Test Coverage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Connections</td>
<td>enatest searches for all the active and inactive connections between the host and the enclosure and reports the number of existing active connections. If the VERBOSE mode is enabled, the port on the host side and the GBIC port on the enclosure side is reported for each active connection. The test also diagnoses any inactive connection(s) and reports the possible causes for the failure. The test will fail if there are one or more inactive connections. See the section on “enatest Fault Isolation Capability” on page 112 for more information.</td>
</tr>
</tbody>
</table>
Sample Output, for an enclosure attached to an SBus socal card:

SUNWvts.enatest.1010 06/05/97 13:48:53 enatest ses0 VERBOSE:
"MYBOX: Lower-Right GBIC connected to host via
/devices/sbus@1f,0/SUNW,socalsoc@0,0:1"
SUNWvts.enatest.1006 06/05/97 13:48:53 enatest ses0 VERBOSE:
"MYBOX: Interface Board (Bottom one in the enclosure) detected to be
installed
and OK"
SUNWvts.enatest.6023 06/05/97 13:48:53 enatest ses0
ERROR: "MYBOX: Cannot communicate with the enclosure via
/devices/sbus@1f,0/SUNW,socalsoc@0,0:0; possibly connected to Lower-Left
GBIC in the enclosure"
Probable_Cause(s):
(1)Signal too low at the GBIC module in the enclosure
(2)Faulty cable or cable disconnected
(3)Faulty GBIC module on the host side
Recommended_Action(s):
(1)Ensure the cables are properly connected
(2)Please contact your service representative
SUNWvts.enatest.2006 06/05/97 13:48:53 enatest ses0 INFO:
"MYBOX: Number of connections to the host: 1"

Disk Access
During the testing, each disk is accessed through each active
connection leading to that disk. The enatest opens partition 2 on
the disk and reads 512 bytes of raw data. If there are any failures,
the test tries to isolate the fault to either an enclosure element, the
cable, the host adapter card, or the OE module on the host adapter.
See the section on "enatest Fault Isolation Capability" on page 112
for more information.

Enclosure Status
The status of the enclosure is obtained by querying the SCSI
Enclosure Services (SES) device in the enclosure. Detailed
information regarding the status of the elements within the
enclosure is reported. The test fails if a critical condition is detected
in the enclosure. The table below shows how the status information
is reported.
TABLE 16-2  Element Enclosure Status

<table>
<thead>
<tr>
<th>Enclosure Element</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk</td>
<td>• Fault Sensed—Yes/No</td>
</tr>
<tr>
<td></td>
<td>• Status of ports A and B—Connected or Bypassed</td>
</tr>
<tr>
<td>Power Supply</td>
<td>• Status—ON/OFF</td>
</tr>
<tr>
<td></td>
<td>• Temperature—OK/Critical Overtemp/Abnormal</td>
</tr>
<tr>
<td></td>
<td>• AC Input—OK/Not OK</td>
</tr>
<tr>
<td></td>
<td>• DC Output—OK/Not OK</td>
</tr>
<tr>
<td>Fan</td>
<td>• Status—On/Off</td>
</tr>
<tr>
<td></td>
<td>• Speed—High/Low/Stopped</td>
</tr>
<tr>
<td>Backplane</td>
<td>• Status—OK/Failed</td>
</tr>
<tr>
<td></td>
<td>• Status of ports A and B—Connected/Bypassed</td>
</tr>
<tr>
<td>Interface Board</td>
<td>• Temperature—OK/Critical Overtemp</td>
</tr>
<tr>
<td></td>
<td>• Loop A status—OK/Failed</td>
</tr>
<tr>
<td></td>
<td>• Loop B status—OK/Failed</td>
</tr>
<tr>
<td>GBIC</td>
<td>• Status—Disabled/Enabled</td>
</tr>
<tr>
<td></td>
<td>• Signal Level—OK/Too low</td>
</tr>
<tr>
<td></td>
<td>• Transmitter—OK/Failed</td>
</tr>
</tbody>
</table>

enatest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User's Guide for more details.
FIGURE 16-1 enatest Test Parameter Options Dialog Box
TABLE 16-3  enatest Options

<table>
<thead>
<tr>
<th>enatest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enclosure Services</td>
<td>Certain control operations are performed on devices in the enclosure through the SES device and verified that the operation was performed successful. This functional test involves the following steps:</td>
</tr>
<tr>
<td>Functional test</td>
<td>1. Perform control operation.</td>
</tr>
<tr>
<td>(general description)</td>
<td>2. Verify control operation was successful.</td>
</tr>
<tr>
<td></td>
<td>3. Restore state to what it was before 1.</td>
</tr>
<tr>
<td></td>
<td>4. Verify restore operation was successful.</td>
</tr>
<tr>
<td></td>
<td>The test will fail if any one of the above steps fails.</td>
</tr>
<tr>
<td></td>
<td>This test targets the disks and the fans in the enclosure.</td>
</tr>
<tr>
<td>Enclosure Services</td>
<td>1. Control Operation—Each port of the disk is toggled from its original state. A port that was originally connected will be bypassed and vice-versa.</td>
</tr>
<tr>
<td>Functional test</td>
<td>2. Verify Control Operation—This is done in two ways. First, the new status of the disk ports is verified by reading the status through the SES device. The test will fail if the status read back does not reflect the change. Next, disk access is attempted through the port that was originally connected but has now been bypassed. The test will fail if the access attempt is successful.</td>
</tr>
<tr>
<td>(detailed Disk test description)</td>
<td>3. Restore State—The port states are restored to what they were before the Control Operation.</td>
</tr>
<tr>
<td></td>
<td>4. Verify Restore Operation—This is done in two ways. First the status of the disk ports is verified by reading the status through the SES device. The test will fail if the status read back does not reflect the change. Next, disk access is attempted through the port that have been reconnected. The test will fail if the access attempt is unsuccessful.</td>
</tr>
<tr>
<td>Enclosure Services</td>
<td>1. Control Operation—The speed of each fan is toggled. Possible speeds are HIGH and LOW.</td>
</tr>
<tr>
<td>Functional test</td>
<td>2. Verify Control Operation—The status is read back through the SES device and the speeds are compared. Failure to vary the speed in this case will result in an INFO message indicating that the fan speed could not be changed but does not result in a test failure. This is because SES can ignore fan speed change requests if required because of existing ambient temperature conditions.</td>
</tr>
<tr>
<td>(detailed Fan test description)</td>
<td>3. Restore State—Restore the fan speed to the original speed.</td>
</tr>
<tr>
<td></td>
<td>4. Verify Restore Operation—This is similar to the Verify Control Operation step above.</td>
</tr>
</tbody>
</table>
enatest Fault Isolation Capability

In the case of a failure, the test aids in fault isolation by reporting the possible cause(s) of failure. The fault isolation capability varies depending on the nature of the fault and the system configuration. enatest can detect and isolate hard faults. The following table shows the fault isolation capability for different configurations. A Yes indicates that fault isolation capability is available for that component in that configuration and a No indicates lack of fault isolation capability for that component in that configuration.

The following table is not applicable when using the PCI-based Fibre Channel card due to the card’s lack of fault isolation capabilities.

**TABLE 16-4 enatest Fault Isolation Configurations**

<table>
<thead>
<tr>
<th>Connections to Enclosure</th>
<th>System Architecture</th>
<th>SOC+ Host Adapter</th>
<th>Host Side GBIC or Cable</th>
<th>Enclosure Elements</th>
<th>IB</th>
<th>GBIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple</td>
<td>sun4u</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Multiple</td>
<td>sun4d</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Single</td>
<td>sun4u</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Single</td>
<td>sun4d</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
**enatest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection test</td>
<td>Yes</td>
<td>In this mode, the host connections and the status of the enclosure are checked. The test fails if there are any broken connections or if a critical enclosure condition is detected. Noncritical conditions result in a warning. A sample of the output follows.</td>
</tr>
</tbody>
</table>

Connection test starting....

ses0
Status: Connected
Enclosure:
   Product Anemones Enterprise Network Array,
   Enclosure Name=MYBOX,
Host Connections:
   Number of Active Connections=2,
Enclosure State:
   Critical Conditions=None, Non-Critical Conditions=None

Connection test complete

<table>
<thead>
<tr>
<th>Test Mode (offline)</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>All test options are allowed in this mode.</td>
</tr>
</tbody>
</table>

**enatest Command-Line Syntax**

```
/opt/SUNWvts/bin/enatest standard_arguments -o dev=device_name, disk_access=enable|disable,disks=disk1:disk2:disk3:...diskn,disp=enable|disable,
```
esfunc=enable|disable, conn=enable|disable, delay=delay_in_seconds

TABLE 16-6  enatest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the name of an ses device in the enclosure.</td>
</tr>
<tr>
<td>disk_access=enable</td>
<td>Enables or disables disk access.</td>
</tr>
<tr>
<td>disks=disk1:disk2:disk3:...</td>
<td>Lists the disks contained in this enclosure that the disk_access test should attempt to access. The disk names are separated by colons (:). If this option is disabled, the test will access all disks found in the enclosure.</td>
</tr>
<tr>
<td>disp=enable</td>
<td>Enables or disables the display of detailed status information regarding the enclosure elements.</td>
</tr>
<tr>
<td>esfunc=enable</td>
<td>Enables or disables the Enclosure Services Functional test.</td>
</tr>
<tr>
<td>conn=enable</td>
<td>Displays information about the connection to the host.</td>
</tr>
<tr>
<td>delay=delay_in_seconds</td>
<td>Sets the minimum delay (in seconds) between successive invocations of the test.</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Sun StorEdge 1000 Enclosure Test (enctest)

enctest tests the Sun StorEdge™ A1000, D1000, and D2 disk enclosures. The A1000 and D1000 enclosures can support either 12 1” 4GB drives or 8 1.6” 9GB drives. The D2 disk enclosure can support 12 1” either 18GB or 36GB drives. These disk enclosures have redundant power and cooling. Following enclosure models are supported by enctest:

- Sun StorEdge A1000—Disk tray with the hardware RAID controller
- Sun StorEdge D1000—Disk tray without the hardware RAID controller
- Sun StorEdge D2—Disk tray without the hardware RAID controller

enctest can be used for validation, configuration verification, repair verification, and fault isolation of these models.

enctest sets the disk enclosure LEDs as follows:

<table>
<thead>
<tr>
<th>LED State</th>
<th>Indicated Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>On</td>
<td>Unrecoverable or critical condition</td>
</tr>
<tr>
<td>Blinking</td>
<td>Non-critical condition</td>
</tr>
<tr>
<td>Off</td>
<td>Informational or no condition to report</td>
</tr>
</tbody>
</table>

**Note** – The ses driver must be present on the system with the Sun StorEdge enclosure(s). If the ses driver is not present, SunVTS will not detect the Sun StorEdge hardware. Use the following command to see if the ses driver is present: `pkginfo SUNWses`

Install the package if necessary.
enctest reports the status of the various elements in the enclosure. An error is registered if an unrecoverable or critical condition is detected. Noncritical conditions are reported through warning messages. The table below describes the information that is reported for each enclosure element.

<table>
<thead>
<tr>
<th>Enclosure Element</th>
<th>Information reported (per device slot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk</td>
<td>Present/Not Present</td>
</tr>
<tr>
<td></td>
<td>Failed/OK</td>
</tr>
<tr>
<td>Power supply</td>
<td>Present/Not Present</td>
</tr>
<tr>
<td></td>
<td>Failed/OK</td>
</tr>
<tr>
<td>Fan</td>
<td>Present/Not Present</td>
</tr>
<tr>
<td></td>
<td>Failed/OK</td>
</tr>
<tr>
<td>Temperature</td>
<td>OK/Over temperature</td>
</tr>
<tr>
<td>RPA cache Battery</td>
<td>OK/Low Charge</td>
</tr>
<tr>
<td>(StorEdge A1000 only)</td>
<td></td>
</tr>
</tbody>
</table>

enctest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 17-1 enctest Test Parameter Options Dialog Box

TABLE 17-3 enctest Options

<table>
<thead>
<tr>
<th>enctest</th>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Display Enclosure Status</td>
<td>When enabled, the status of the enclosure elements are displayed as messages in the SunVTS console. These messages are also logged in the SunVTS info log file.</td>
</tr>
<tr>
<td></td>
<td>LED Test</td>
<td>When enabled, the LEDs associated with the disks are flashed for a short period of time.</td>
</tr>
<tr>
<td></td>
<td>Interval Between Passes</td>
<td>Sets the minimum delay between successive invocations of the test.</td>
</tr>
</tbody>
</table>
enctest Test Modes

TABLE 17-4 enctest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connectivity</td>
<td>Yes</td>
<td>The enclosure status summary bits are read. Detailed information is reported only if an unrecoverable or critical condition is detected.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>All options are allowed.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

enctest Command-Line Syntax

```
/opt/SUNWvts/bin/enctest standard_arguments -o dev=device_name, disp=enable|disable, led=enable|disable, delay=delay_in_seconds
```

TABLE 17-5 enctest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>This is the name of a ses device in the enclosure.</td>
</tr>
<tr>
<td>disp=enable</td>
<td>disable</td>
</tr>
<tr>
<td>led=enable</td>
<td>disable</td>
</tr>
<tr>
<td>delay=delay_in_seconds</td>
<td>Minimum delay between successive invocations of the test.</td>
</tr>
</tbody>
</table>

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 “32-Bit and 64-Bit Tests” on page 13.
Environmental Test (envtest)

envtest exercises the I2C bus on the Sun Enterprise™ 450 product line. envtest contains five subtests to test and report on the power supply status, system temperature status, fan speed, disk LEDs, and front panel and keyswitch.

envtest is not scalable.

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

envtest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 18-1 envtest Test Parameter Options Dialog Box
<table>
<thead>
<tr>
<th>envtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk LEDs test</td>
<td>Illuminates each LED on the disk backplane(s) to green, then amber, and then back to its original state. The test then illuminates all disk LEDs to green, then amber, and then back to their original state. This test is only enabled in Functional test mode.</td>
</tr>
<tr>
<td>Fan test</td>
<td>Cycles each fanbank speed to low, medium, and high, then verifies the correct speed. Next, each fanbank is stopped, one at a time. The test then verifies that a fan fault has occurred. Next, the watchdog timer is invoked to simulate a catastrophic failure. The test verifies that the system set all fanbanks to high and then resets the fan speed to normal. This test is only enabled in Functional test.</td>
</tr>
<tr>
<td>Front Panel and Keyswitch test</td>
<td>Flashes each individual LED on the front panel to ON (green or amber), then OFF, and then back to its original state. The test then illuminates all front panel LEDs then sets them back to their original state. The power on LED is Read Only and will not be cycled. The test then displays the current keyswitch position. This test is only enabled in Functional test mode.</td>
</tr>
<tr>
<td>Power Supply Status</td>
<td>Identifies the number of power supplies that are in the system, and the state of each power supply, and verifies that the power supply temperatures are within normal operating parameters. This test is enabled in all modes.</td>
</tr>
<tr>
<td>Temperature Status</td>
<td>Identifies the current temperature of each CPU in the system, and the ambient temperature of the system, and envtest verifies that all temperatures are within normal operating parameters. This test is enabled in all modes.</td>
</tr>
</tbody>
</table>
envtest Test Modes

envtest supports Connection and Functional tests.

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Reports the status of the power supplies, the temperature sensors within the system, and verifies normal operating parameters.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Tests the disk back panel, front panel LEDs, and fan control circuitry. Also uses the same functionality as online Functional mode and connection mode</td>
</tr>
</tbody>
</table>

envtest Command-Line Syntax

```
/opt/SUNWvts/bin/envtest [standard arguments]
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dev=raw_device_name</code></td>
<td>Specifies the name of the raw device to test.</td>
</tr>
<tr>
<td>`diskleds=enable</td>
<td>disable`</td>
</tr>
<tr>
<td>`fans=enable</td>
<td>disable`</td>
</tr>
<tr>
<td>`fpanel=enable</td>
<td>disable`</td>
</tr>
<tr>
<td>`psupply=enable</td>
<td>disable`</td>
</tr>
<tr>
<td>`temp=enable</td>
<td>disable`</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Environmental Test (env2test)

e2nv test exercises and validates the I2C bus on the Sun Enterprise 250 systems.

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

env2test is not scalable.

Note – Do not run env2test while the system is under a heavy load or false failures may be reported.

env2test Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 19-1 env2test Test Parameter Options Dialog Box
**env2test Options**

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

**env2test Test Modes**

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

**TABLE 19-2  env2test Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Reports the status of the power supplies and the temperature sensors within the system, and verifies normal operating parameters</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Tests the disk back panel, front panel LEDs, and fan control circuitry. Also uses the same functionality as online mode and connection test</td>
</tr>
</tbody>
</table>
env2test Command-Line Syntax

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

TABLE 19-3 env2test Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=raw_device_name</td>
<td>Specifies the name of the raw device to test.</td>
</tr>
<tr>
<td>diskleds=enable</td>
<td>disable</td>
</tr>
<tr>
<td>env_mon=poll_interval</td>
<td>Displays all system environmental statics every poll_interval seconds. (Display ONLY, does not test.)</td>
</tr>
<tr>
<td>fans=enable</td>
<td>disable</td>
</tr>
<tr>
<td>fpanel=enable</td>
<td>disable</td>
</tr>
<tr>
<td>psupply=enable</td>
<td>disable</td>
</tr>
<tr>
<td>temp=enable</td>
<td>disable</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Environmental Test (env3test)

env3test is an environmental control test for the Sun Blade 1000 and Sun Blade 100 systems. The test monitors the system by reading temperatures and fan speeds, as well as their limits. It reports whether the temperatures and fan speeds fall within system environmental condition limits.

The MAX1617 temperature sensor keeps a set of hard limits for the maximum and minimum temperatures allowed within the system. If the temperature passes one of these limits, the system performs a hard shutdown to protect hardware components. env3test also checks against the environmental monitor daemon, which keeps its own limits.

By reading these temperature values, the test reports the possibility of a failure of the system environmental feedback loop between the fans and the temperature sensors.

**Note** – If env3test fails to register temperature values, the system temperature indicators may be faulty.

**Note** – env3test will not run on operating environments earlier than the Solaris 8 10/00 operating environment.

env3test 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.
If the Log File option is set to True, the test logs two lines of information read from the system into the log file `/var/opt/SUNWvts/logs/env3test.log`. The first line is a time stamp. The second line is a list of names and the read values, as shown below:

```
system-fan, 19, cpu-fan, 49, power-supply-fan, 100, cpu, 81, cpu-ambient, 24
```

The values for fan settings refer to the percentage of performance at which each fan is running. The system fan, for example, is running at 19% of its capacity. The temperature values are in degrees Celsius. The cpu in the above example is running at 81 degrees C.
**env3test Test Modes**

**TABLE 20-1 env3test Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Attempts connection to the device.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Reports the received information to the GUI logging window in verbose mode.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**env3test Command-Line Syntax**

```
/opt/SUNWvts/bin/env3test [standard arguments] -o dev=driver_name, logging=true|false
```

**TABLE 20-2 env3test Command-Line Syntax**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=driver_name</td>
<td>Specifies the driver name for this test. The driver for the Sun Blade 1000 is max1617.</td>
</tr>
<tr>
<td>logging=true</td>
<td>false</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Environmental Test (env4test)

env4test exercises and validates the I2C bus on the Sun Netra™ CT systems.

There are nineteen subtests in env4test that test and report the status of the following:

- Continuity LED flash test
- Slot LED flash test
- Disk LED flash test
- Power supply LED flash test
- Fan LED flash test
- System controller board LED flash test
- Front transition module LED flash test
- Green LED’s blink register
- Register read/write test, 3N march
- Register read/write test, 5N march
- Register read/write test, Crosstalk
- FRU presence test
- I/O slot health test
- I/O slot reset test
- Power remote on/off test
- Fan speed switch test
- CPU temperature status display test
- Power supply status display test
- Fan status display test

env4test is not scalable.
env4test Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 21-1 env4test Test Parameter Options Dialog Box
<table>
<thead>
<tr>
<th>env4test Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEDS ON Continuity test</td>
<td>Asserts all LEDs in the system display panel except for the PDU LEDs. This verifies the ability of the System Controller Board (SCB) to perform sequential writes while automatically advancing the address pointer internally.</td>
</tr>
<tr>
<td>Slot LEDS test</td>
<td>Repeatedly flashes all I/O slot LEDs.</td>
</tr>
<tr>
<td>Disk LEDS test</td>
<td>Repeatedly flashes all disk LEDs.</td>
</tr>
<tr>
<td>Power Supply LEDS test</td>
<td>Repeatedly flashes all power supply LEDs.</td>
</tr>
<tr>
<td>Fan LEDS test</td>
<td>Repeatedly flashes all fan LEDs.</td>
</tr>
<tr>
<td>SCB LEDS test</td>
<td>Repeatedly flashes all system controller board LEDs.</td>
</tr>
<tr>
<td>FTM LEDS test</td>
<td>Repeatedly flashes all front transition module LEDs.</td>
</tr>
<tr>
<td>Green LED Blink test</td>
<td>Tests hardware functionality that enables blinking for the green LEDs.</td>
</tr>
<tr>
<td>Register R/W test</td>
<td>A pattern write test to interrupt registers and LED registers (read/write) to isolate the stuck-at-1, stuck-at-0, and cross talk fault categories. The test execution time from shortest to longest is: 3N march, 5N march, (XTALK) Cross Talk.</td>
</tr>
<tr>
<td>FRU Presence test</td>
<td>Reads and displays FRU presence signals.</td>
</tr>
<tr>
<td>Health test</td>
<td>Reads and displays I/O slot health signals.</td>
</tr>
<tr>
<td>Reset test</td>
<td>Sequentially resets individual I/O slots.</td>
</tr>
<tr>
<td>Power Supply test</td>
<td>Tests the remote power on/off signals in systems with redundant power supplies. This test is only valid when 2 power supplies are inserted in a single drawer.</td>
</tr>
<tr>
<td>Fan test</td>
<td>Tests the functionality of fan speed changes.</td>
</tr>
<tr>
<td>Temperature Status test</td>
<td>Identifies the current temperature of the CPU, the ambient temperature of the system, and verifies that the temperature is within normal operating parameters. This test is enabled in all modes.</td>
</tr>
<tr>
<td>Power Supply Status test</td>
<td>Detects assertion of Presence signal in the SCB and reads the power supply status register via the I2C bus. The encoded bit status is displayed to the console.</td>
</tr>
<tr>
<td>Fan Status test</td>
<td>Detects assertion of Presence signal in the SCB and reads the fan status register via the I2C bus. The encoded bit status is displayed to the console.</td>
</tr>
<tr>
<td>All tests</td>
<td>Executes all tests sequentially.</td>
</tr>
</tbody>
</table>


env4test Test Modes

env4test supports Connection and Functional tests as described in the table below.

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Tests the disk back panel, front panel LEDs, and fan control circuitry. Also uses the same functionality as Online mode and Connection test mode.</td>
</tr>
</tbody>
</table>

env4test Command-Line Syntax

/opt/SUNWvts/bin/env4test [standard arguments]  -o dev=device_name, ContinuityTest=E(nable)|D(isable), SlotLedsTest=E|D, DiskLedsTest=E|D, PsupplyLedsTest=E|D, FanLedsTest=E|D, ScbLedsTest=E|D, FtmLedsTest=E|D, GreenLedsBlinkTest=E|D, RegTest=3N|5N|Cross_Talk, FRUPresenceTest=E|D, HealthTest=E|D, ResetTest=E|D, PowerSupplyTest=E|D, FanTest=E|D, TempStatus=E|D, PsupplyStatus=E|D, FanStatus=E|D, AllTests=E|D

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=raw_device_name</td>
<td>Specifies the name of the raw device to test.</td>
</tr>
<tr>
<td>ContinuityTest=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>SlotLedsTest=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>DiskLedsTest=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>PsupplyLedsTest=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>FanLedsTest=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>ScbLedsTest=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>FtmLedsTest=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>GreenLedsBlinkTest=</td>
<td>Enables or disables the green LED hardware blink test.</td>
</tr>
</tbody>
</table>
### TABLE 21-3  env4test Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegTest=3N</td>
<td>5N</td>
</tr>
<tr>
<td>FRUPresenceTest= Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>HealthTest= Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>ResetTest= Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>PowerSupplyTest= Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>FanTest=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>TempStatus=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>PsupplyStatus=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>FanStatus=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>AllTests=Enable</td>
<td>Disable</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Environmental Test (env5test)

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

This test is not scalable.

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

---

**env5test Test Requirements**

- You must install the SUNWpiclh, SUNWpiclr, SUNWpiclu, and SUNWpiclx picl packages correctly before running env5test.
- Verify that the picld daemon is running by typing the following:

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

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

```bash
#/etc/init.d/picld start
```
env5test Options

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

FIGURE 22-1 env5test Test Parameter Options Dialog Box
<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fan test</td>
<td>Cycles each fan bank in the system and identifies its current speed and state. Displays fault information if the state is not correct. Provides different test coverage for various types of fans based on their properties. Only enabled in Functional test mode.</td>
</tr>
<tr>
<td>Keyswitch status</td>
<td>Displays the current keyswitch position. Enabled in all modes.</td>
</tr>
<tr>
<td>LEDs test</td>
<td>Enables or disables the LEDS subtest. Default is Disable. Flashes each individual green or amber LED in the system on, then off, then back to its original state. You can select to test all LEDs (by default) or test by categories. Also specifies how long the LEDs stay on during the exercise. Only enabled in Functional test mode.</td>
</tr>
<tr>
<td>LED Type</td>
<td>Sets the type of LED category to be tested. Default is All.</td>
</tr>
<tr>
<td>LED hold time</td>
<td>Sets the number of seconds that LEDs are turned on during the LED subtest. Values are 0 to 10. Default is 0.</td>
</tr>
<tr>
<td>Power supply status</td>
<td>Identifies the number of power supplies in the system, the state of each one, and current (I) draw information if applicable. Displays fault information if the state of the power supply is incorrect. Enabled in all modes.</td>
</tr>
<tr>
<td>Temperature status</td>
<td>Identifies the current temperature of temperature sensors in the system and verifies that all temperatures are within normal operating parameters. Enabled in all modes.</td>
</tr>
</tbody>
</table>
env5test Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Runs the Keyswitch, Power Supply, and Temperature subtests. Reports on the status only of system fans. Verifies normal operating parameters.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of subtests.</td>
</tr>
</tbody>
</table>

env5test Command-Line Syntax

```
/opt/SUNWvts/bin/sparcv9/env5test standard_arguments  
-o dev=raw_device_name,leds=E|D,ledtype=category,ledtime=num_seconds,  
keys=E|D,fans=E|D,psupply=E|D,temp=E|D
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=raw_device_name</td>
<td>Specifies the name of the raw device to test. Default is /dev/env</td>
</tr>
<tr>
<td>leds=E</td>
<td>D</td>
</tr>
<tr>
<td>ledtype=category</td>
<td>Sets the type of LED category to be tested. Default is All.</td>
</tr>
<tr>
<td>ledtime=num_seconds</td>
<td>Sets the number of seconds that LEDs are turned on during the LED subtest. Values are 0 to 10. Default is 0.</td>
</tr>
<tr>
<td>keys=E</td>
<td>D</td>
</tr>
<tr>
<td>fans=E</td>
<td>D</td>
</tr>
<tr>
<td>psupply=E</td>
<td>D</td>
</tr>
<tr>
<td>temp=E</td>
<td>D</td>
</tr>
</tbody>
</table>
Frame Buffer Test (fbtest)

fbtest is a generic test for all frame buffers used with the Solaris 2.x and Solaris 7-9 software.

The fbtest checks the frame buffer by sequentially writing, reading, and verifying small blocks of random patterns across the entire video RAM. The block size is 64 x 64 pixels. If a miscompare occurs, the test stops with an error message that indicates the location of the error.

If a generic frame buffer device name (dvc/fb) is specified, fbtest automatically detects the depth of the frame buffer, and adjusts testing to the frame buffer size.

For full instructions on testing frame buffers, see “Testing Frame Buffers” on page 18.

fbtest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
**FIGURE 23-1** fbttest Test Parameter Options Dialog Box

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

Due to the nature of graphic tests, reading from or writing to the frame buffer during graphic tests disturbs user operation.

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full set of tests</td>
</tr>
</tbody>
</table>

fbtest Command-Line Syntax

`/opt/SUNWvts/bin/fbtest` `standard_arguments -o dev=device_name, lock=Enable/Disable`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies which frame buffer to test.</td>
</tr>
<tr>
<td>lock=Enable/Disable</td>
<td>Enables or disables the window system locking option. See “Testing Frame Buffers” on page 18 for details. Frame buffer locking is enabled by default on the window server running the Open Windows software.</td>
</tr>
</tbody>
</table>

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. For more information refer to “32-Bit and 64-Bit Tests” on page 13.
Fast Frame Buffer Test (*ffbtest*)

*ffbtest* verifies the functionality of the fast frame buffer (FFB).

*ffbtest* can detect and adapt to the video modes of single- and double-buffer versions of the FFB. All *ffbtest* tests can run in several screen resolutions such as standard, stereo, and high resolution. In stereo mode, all tests write into the right and left eyes unless you specify otherwise. This test also supports FFB, FFB2, and FFB2+ fast frame buffer configurations. Use the *ffbconfig* `-prconf` command to display the configuration of the frame buffer you want to test.

You can interrupt *ffbtest* using Control-C.

Test accuracy is checked using a checksum algorithm. Possible locations of failing pixels are identified, as well as the likely failing FRU.

---

**Caution – Do not** run any other application or screen saver program that uses the FFB accelerator port while running *ffbtest*. These programs cause SunVTS to return incorrect errors.

---

*ffbtest* Test Requirements

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

For full instructions on testing frame buffers, see “Testing Frame Buffers” on page 18.

*ffbtest* requires approximately 7 MB of disk space in the `/tmp` directory to extract its working files. If this space is not available, the diagnostic will fail and report warning and error messages, indicating a lack of disk space.
To start SunVTS with vtsui, but without vtsk, you must add the host name to xhost as: \texttt{xhost + <hostname>}

\section*{ffbtest 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 \textit{SunVTS User's Guide} for more details.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{ffbtest_test_options_dialog.png}
\caption{ffbtest Test Parameter Options Dialog Box}
\end{figure}
By default, all tests are enabled except the stereo test.

**TABLE 24-1  ffbtest Options**

<table>
<thead>
<tr>
<th>ffbtest Options</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3DRAM test      | The 3DRAM test thoroughly tests the video memory in the FFB using 512-bit reads and writes. 3DRAM makes a full screen pass, consisting of a write and a read to each pixel location, for each access mode on the list below. You can use either random data or specify data at the command line. A second pass is made with the one’s complement of the data used in the first pass so that each memory location is tested with both a zero and a one. Notice that some passes are skipped on the single-buffered FFB. Errors in this subtest are attributes to the 3DRAM. A failing chip is indicated by (X, Y) locations and device-specific “U” numbers:
  • DFB8R, DFB8G, DFB8B, DFB8X—Buffer A
  • DFB24—Buffer A
  • DFB32—Buffer A
  • SFB8R, SFB8G, SFB8B, SFB8X—Buffer A
  • SFB8R, SFB8G, SFB8B, SFB8X—Buffer B (double buffer only)
  • SFB32—Buffer A
  • SFB32—Buffer B (double buffer only)
  • SFB32—Buffer C (double buffer only)
  • SFB64—Buffers A and C (double buffer only)
  • SFB64—Buffers B and C (double buffer only) |
3DRAM Logic test

3DRAM Logic provides logical functionality to the FFB. The following services are tested:

- Compare Controls—Match AB
- Compare Controls—Magnitude AB
- Compare Controls—Match C (double buffer only)
- Compare Controls—Magnitude C (double buffer only)
- Match Mask—AB
- Magnitude Mask—AB
- Match Mask—C (double buffer only)
- Magnitude Mask—C (double buffer only)
- Raster Operations—RGB
- Raster Operations—X
- Raster Operations—YZ (double buffer only)
- Plane Mask—RGB
- Plane Mask—X
- Plane Mask—Y
- Plane Mask—Z
- Group Enable—R, G, B, X
- Group Enable—Y, Z (double buffer only)

ffbtest tests each function separately with a series of SFB64 writes. A total of 16 writes are made for each different test case with Y coordinate values varying from 0 to 30 in increments of 2 pixels. This dotted column organization provides page thrashing and block flashing in all screen resolutions. For each operation, all possible combinations are tested. For example, \(ROP\) RGB new==old has three possible values: new < old, new == old, and new > old. \(ffbtest\) tests each of these cases.

Five passes of the functions are made. Each pass writes into a different FFB address space: SFB32-A, SFB32-B, SFB32-C, SFB64-AC, and SFB64-BC. Note that the passes that write into the SFB32 address spaces are writing two pixels at a time because the tests use SFB64 writes.

For FFB2+ boards, additional testing is performed on the new stencil and passin capabilities if the board is DBZ.

Care is taken to ensure that all 3DRAM chips are tested. Errors in this subtest are attributed to the 3DRAM.
RAMDAC Test

RAMDAC registers are tested using simple read/write patterns to
determine if there are any bad bits. This includes all LUTs.
*ffbtest* ensures that data is actually being read from the
RAMDAC and not being supplied by the driver.

Next, the RAMDAC Signature Register captures the pixels going to
the screen. This test determines that all of the different data paths
within the RAMDAC are functioning properly.

The following modes are tested:
- 24-bit true color from A
- 24-bit true linear color from A
- 24-bit direct color from A
- 24-bit true color from B (double buffer only)
- 24-bit true linear color from B
- 24-bit direct color from B (double buffer only)
- 8-bit pseudo color (from each plane in RGB) from A
- 8-bit pseudo color (from each plane in RGB) from B (double
  buffer only)
- 8-bit non-linear grayscale (from each plane in RGB) from A
- 8-bit non-linear grayscale (from each plane in RGB) from B
  (double buffer only)
- 8-bit linear grayscale (from each plane in XRGB) from A
- 8-bit linear grayscale (from each plane in XRGB) from B (double
  buffer only)
- 8-bit overlay pseudo color (from buffer A, X plane)

This test displays a total of 11 different types of windows on the
screen for the single-buffered configuration; 22 for double-
buffered. A cursor is also displayed on the screen.

RAMDAC on FFB2+ board supports three modes (SEP8, SEP4, and
Combined). This test detects the RAMDAC type and tests the
original and additional features like increased number of CLUTs,
increased WLUT size, additional overlay WLUT.

Errors in this test are attributed to the RAMDAC.

**TABLE 24-1  *ffbtest* Options (Continued)**

<table>
<thead>
<tr>
<th><strong>ffbtest Options</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>RAMDAC Test</td>
<td>RAMDAC registers are tested using simple read/write patterns to determine if there are any bad bits. This includes all LUTs. <em>ffbtest</em> ensures that data is actually being read from the RAMDAC and not being supplied by the driver.</td>
</tr>
<tr>
<td></td>
<td>RAMDAC Signature Register captures the pixels going to the screen. This test determines that all of the different data paths within the RAMDAC are functioning properly.</td>
</tr>
<tr>
<td></td>
<td>The following modes are tested:</td>
</tr>
<tr>
<td></td>
<td>• 24-bit true color from A</td>
</tr>
<tr>
<td></td>
<td>• 24-bit true linear color from A</td>
</tr>
<tr>
<td></td>
<td>• 24-bit direct color from A</td>
</tr>
<tr>
<td></td>
<td>• 24-bit true color from B (double buffer only)</td>
</tr>
<tr>
<td></td>
<td>• 24-bit true linear color from B</td>
</tr>
<tr>
<td></td>
<td>• 24-bit direct color from B (double buffer only)</td>
</tr>
<tr>
<td></td>
<td>• 8-bit pseudo color (from each plane in RGB) from A</td>
</tr>
<tr>
<td></td>
<td>• 8-bit pseudo color (from each plane in RGB) from B (double buffer only)</td>
</tr>
<tr>
<td></td>
<td>• 8-bit non-linear grayscale (from each plane in RGB) from A</td>
</tr>
<tr>
<td></td>
<td>• 8-bit non-linear grayscale (from each plane in RGB) from B (double buffer only)</td>
</tr>
<tr>
<td></td>
<td>• 8-bit linear grayscale (from each plane in XRGB) from A</td>
</tr>
<tr>
<td></td>
<td>• 8-bit linear grayscale (from each plane in XRGB) from B (double buffer only)</td>
</tr>
<tr>
<td></td>
<td>• 8-bit overlay pseudo color (from buffer A, X plane)</td>
</tr>
<tr>
<td>RAMDAC test</td>
<td>This test displays a total of 11 different types of windows on the screen for the single-buffered configuration; 22 for double-buffered. A cursor is also displayed on the screen.</td>
</tr>
<tr>
<td>(Continued)</td>
<td>RAMDAC on FFB2+ board supports three modes (SEP8, SEP4, and Combined). This test detects the RAMDAC type and tests the original and additional features like increased number of CLUTs, increased WLUT size, additional overlay WLUT.</td>
</tr>
<tr>
<td></td>
<td>Errors in this test are attributed to the RAMDAC.</td>
</tr>
</tbody>
</table>
Rendering Pipeline

The Rendering Pipeline uses the rendering pipeline tests developed for the FFB stand-alone diagnostics. Each primitive is tested thoroughly with the following sources and configurations:

- Dots
- Anti-aliased dots
- Lines using all four line drawing primitives
- Triangles
- Polygons
- Rectangles
- Fonts

Errors in this test are attributed to the FBC.

Fast Fill/Vertical Scroll test

The Fast Fill/Vertical Scroll primitives are separated from the Rendering Pipeline tests because of their dependence on screen type. There are three different tests, one for each screen type. Each test uses both block and page mode fistfuls.

Errors in this test are attributed to the FBC.

Pixel Process test

Pixel Processor, a subtest, exercises the following options selected by the FFB's Pixel Processor Control (PPC) register:

- Auxiliary clipping (additive and subtractive)
- Depth cueing
- Alpha blend
- Viewport clip (2D and 3D)
- Area pattern (transparent and opaque)

Errors in this test are attributed to the FBC.

Picking test

The Picking test exercises the pick detect login of the 3DRAM. ffbtest defines a pick detect window and checks that writes to the window are picked, and writes outside the window are not picked. The test is repeated once for each 3DRAM.

Errors in this test are attributed to the 3DRAM.
Arbitration test

The Arbitration subtest continuously renders an object into the accelerator port while performing reads and writes through the direct port. For single-buffered configurations, a picture is rendered into the RGB planes while another process does DFB reads and writes in the X plane. For doubled buffered configuration, a picture is rendered into all 32 planes of the B buffer while the other does 32-bit DFB reads and writes in the A plane. This subtest simulates conditions in the real world, where rendering processes and windows operations run concurrently.

Errors in this test are attributed to the FBC.

Stereo Test

The Stereo test displays an object in stereo mode with different images for the right and left eye. The user can verify proper operation by looking at the screen with stereo glasses and following the displayed instructions. If the monitor type is not 1280x1024 at 76MHz, this test prints a warning message and does not run. To prevent this message from being displayed or written to the SunVTS information log, disable the stereo test in the test option menu. Only Sony P4 and N2 monitors support stereo resolutions.

This test temporarily switches the monitor into stereo mode, renders a stereo image, performs a signature analysis on the stereo image (using the RAMDAC signature capture register), and after displaying the image for five seconds, restores the monitor to its previous resolution.

Errors in this test are attributed to the RAMDAC.

Note — If vertical lines are displayed on the console when running SunVTS, this could be caused by the ffbtest stereo test. There is a time-critical period in the FFB hardware when trying to change the screen resolution from standard to stereo and back to standard. When the system is heavily loaded or running all of the SunVTS tests, the FFB device driver may get interrupted while changing screen resolution. If this occurs, FB ASIC and RAMDAC get out of synchronization, resulting in an unusual display on the FFB screen. This problem could also cause a system hang condition. To avoid this type of display problem, disable the ffbtest stereo test when other SunVTS tests are enabled.

This test is disabled by default because it is only needed when a stereo monitor and stereo glasses are present.
ffbtest Test Modes

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

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>The ffbtest verifies both the single- (SFB) and double-buffered (DBZ) fast frame buffer boards.</td>
</tr>
<tr>
<td>Stress mode</td>
<td>Yes</td>
<td>Stress mode exercises the frame buffer as much as possible. The Random test generator, constructed as part of the verification effort, is used. Starting from a known seed, random primitives with random attributes are generated. The primitives are checked to ensure that they were rendered in the same way as on a known good system. The test is repeated ten times, with each random picture overlaying the previous one.</td>
</tr>
</tbody>
</table>

Stress mode is not available on FFB2+ boards.
ffbtest Command-Line Syntax

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

**TABLE 24-3  ffbtest Command-Line Syntax**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td><em>device_name</em> is the relative path name of the device being tested with respect to /dev/fbs; The default is ffb0.</td>
</tr>
<tr>
<td>S=subtest_number</td>
<td><em>subtest_number</em> is the test number of the subtest to be run. Select from the subtests below. You can run multiple subtests by adding the subtest numbers. For example, n=0x3 runs both test 1 and test 2; n=0x180 runs both test 0x080 and test 0x0100. Note that you do not need the leading zeros.</td>
</tr>
<tr>
<td></td>
<td>• n=0x00001 3DRAM</td>
</tr>
<tr>
<td></td>
<td>• n=0x00002 3DRAM Logic</td>
</tr>
<tr>
<td></td>
<td>• n=0x00004 RAMDAC</td>
</tr>
<tr>
<td></td>
<td>• n=0x00008 Rendering Pipeline</td>
</tr>
<tr>
<td></td>
<td>• n=0x00010 FastFill/Vertical Scroll</td>
</tr>
<tr>
<td></td>
<td>• n=0x00020 Pixel Processor</td>
</tr>
<tr>
<td></td>
<td>• n=0x00040 Picking</td>
</tr>
<tr>
<td></td>
<td>• n=0x00080 Arbitration</td>
</tr>
<tr>
<td></td>
<td>• n=0x00100 Stereo</td>
</tr>
<tr>
<td></td>
<td>More than one test can be selected by ORing subtest numbers. For example: n = 0x00009 selects 3DRAM and Rendering Pipeline tests. A hex number must be preceded by 0x, decimal numbers are also acceptable.</td>
</tr>
<tr>
<td>F=#_of_subtest_loops</td>
<td>Specifies the number of times to repeat each subtest. The default is 1.</td>
</tr>
<tr>
<td>B=#_of_test_loops</td>
<td>Specifies the number of times to repeat a test loop before passing. The default is 1.</td>
</tr>
<tr>
<td>P=test_pattern</td>
<td>Specifies the test pattern number. The default is r, for random patterns. You may also choose 0 for 0x0000000, 3 for 0x3333333, 5 for 0x5555555, or 9 for 0x9999999.</td>
</tr>
</tbody>
</table>
**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 “32-Bit and 64-Bit Tests” on page 13.

**Note** – Errors returned by `ffbtest` are nonspecific. It is not possible to determine which component caused a failure. In all error conditions, the field replaceable unit (FRU) is the entire FFB.
Floating Point Unit Test (*fputest*)

*fputest* checks the floating point unit on machines with SPARC architecture.

*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:
- Linpack test
- Cparanoia test
- Kcsqrt test
- Kcdiv test
- Clorenz test
- Cvector test

**fpustest Options**

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.
**FIGURE 25-1** `fputest` Test Parameter Options Dialog Box

**CPU-UNIT0 (`fputest`) Test Parameter Options**

**Configuration:**
Type: SPARC V9 based FPU
Clock-frequency: 167 MHz.

**Options:**

<table>
<thead>
<tr>
<th>Instance:</th>
<th>▲▼ 1</th>
</tr>
</thead>
</table>

**Within Instance:**
Apply

**Across All Instances:**
Apply

Reset  Cancel
fputest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>In this mode, fputest includes all the instruction tests.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs all the instruction tests and all the benchmark tests.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stress mode</td>
<td>Yes</td>
<td>This mode runs several fpu benchmark tests.</td>
</tr>
</tbody>
</table>

fputest Command-Line Syntax

/opt/SUNWvts/bin/fputest standard_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 “32-Bit and 64-Bit Tests” on page 13.
IEEE 1394 Camera Test
(fwcamtest)

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

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

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

fwcamtest Test Requirements

Start a Window Environment

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

**Note** – Your window system must be operating in 24-bit depth to run the display test. Instructions for changing this setting are below.
If you are working in CDE, you can change your system to 24-bit depth by editing the file /usr/dt/config/Xservers or /etc/dt/config/Xservers. The file /etc/dt/config/Xservers overrides the file /usr/dt/config/Xservers. Edit the appropriate file to include the following line:

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

Testing Through a Remote Connection

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

fwcamtest Subtests

fwcamtest has three subtests:

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

fwcamtest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 26-1  fwcamtest Test Parameter Options Dialog Box

TABLE 26-1  fwcamtest Options

<table>
<thead>
<tr>
<th>fwcamtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DcamParamTest</td>
<td>Enables or disables the Parameter subtest.</td>
</tr>
<tr>
<td>FrameRevTest</td>
<td>Enables or disables the Frame Receive subtest.</td>
</tr>
<tr>
<td>DisplayTest</td>
<td>Enables or disables the Display subtest.</td>
</tr>
<tr>
<td>Video Length</td>
<td>Determines, in seconds, how long the screen display lasts.</td>
</tr>
</tbody>
</table>
fwcamtest Test Modes

### TABLE 26-2 fwcamtest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests.</td>
</tr>
</tbody>
</table>

fwcamtest Command-Line Syntax

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

### TABLE 26-3 fwcamtest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=dcam0</td>
<td>Specifies the device name for this test.</td>
</tr>
<tr>
<td>dcamparam=E(nable)</td>
<td>D(isable)</td>
</tr>
<tr>
<td>framercv=E</td>
<td>D</td>
</tr>
<tr>
<td>display=E</td>
<td>D</td>
</tr>
<tr>
<td>T=seconds</td>
<td>Specifies the time period of the display test in seconds.</td>
</tr>
</tbody>
</table>

**Note** – 64-bit tests are located in the sparcv9 subdirectory: `/opt/SUNWvts/bin/sparcv9/testname`. If the test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to “32-Bit and 64-Bit Tests” on page 13.
CHAPTER 27

Graphics Frame Buffer Test
(gfbtest)

**gfbtest** verifies the functionality of the Graphics frame buffer (GFB).

gfbtest can detect and adapt to the video modes of Sun XV-1000 Model D256 Graphics Accelerator and Sun XV-1000 Model S64 Graphics Accelerator. SunXV-1000 Model D256 Graphics Accelerator has 256 MB of Texture Memory and 72 MB of video memory. Sun XV-1000 Model S64 Graphics Accelerator has 36 MB Texture Memory and 256 MB Video Memory.

All gfbtest tests can run in several screen resolutions such as standard, stereo, and high resolution. In stereo mode, all tests write into the right and left eyes unless you specify otherwise. Use the `fbconfig -dev <device-name> -prconf` command to display the configuration of the frame buffer you want to test. You can interrupt gfbtest using Control-C. Turn off all other keyboard input if CDE is running on the unit being tested. Test accuracy is checked using a checksum algorithm. Possible locations of failing pixels are identified, as well as the likely failing FRU.

**Note** – gfbtest is only available in 64-bit mode.

**Caution** – Do not run any other application or screen saver program that uses the GFB accelerator port while running gfbtest. These programs cause SunVTS to return incorrect errors.
gfbtest Test Requirements

Disable all screen savers before testing any graphics device. Type `xset s off` at a UNIX prompt to disable the Solaris screen saver. Disable the Power Management software if it is running. For full instructions on testing frame buffers, see “Testing Frame Buffers” on page 9.

gfbtest requires approximately 26 MB of disk space in the `/tmp` directory to extract its working files. If this space is not available, the diagnostic will fail and report warning and error messages, indicating a lack of disk space. To start SunVTS with `vtsui`, but without `vtsk`, you must add the host name to `xhost` as:

```
xhost + <hostname>
```

---

**gfbtest Options**

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details. By default, all tests are enabled except the stereo test.
FIGURE 27-1 gfbtest Test Parameter Options Dialog Box
### TABLE 27-1  
**gf-test** Options

<table>
<thead>
<tr>
<th><strong>gf-test Options</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| 3DRAM Test          | The 3DRAM test thoroughly tests the video memory in the GFB using 512-bit reads and writes. 3DRAM makes a full screen pass, consisting of a write and a read to each pixel location, for each access mode on the list below. You can use either random data or specify data at the command line. A second pass is made with the one’s complement of the data used in the first pass so that each memory location is tested with both a zero and a one. Errors in this subtest are attributed to the 3DRAM. A failing chip is indicated by (X, Y) locations and device-specific "U" numbers:  
• SFB Stencil 8  
• SFB WID 16  
• FB RGBA 64 - Buffer A  
• SFB RGBA 64 - Buffer B |
3DRAM Logic Test  provides logical functionality to the GFB. The following services are tested:

- Compare Controls - Match AB
- Compare Controls - Magnitude AB
- Compare Controls - Match C
- Compare Controls - Magnitude C
- Match Mask - AB
- Magnitude Mask - AB
- Match Mask - C
- Magnitude Mask - C
- Raster Operations - RGB
- Raster Operations - X
- Raster Operations - YZ
- Plane Mask - RGB

_tests each function separately with a series of SFB64 writes. A total of 16 writes are made for each different test case with Y coordinate values varying from 0 to 30 in increments of 2 pixels. This dotted column organization provides page thrashing and block flashing in all screen resolutions. For each operation, all possible combinations are tested.

For example: \texttt{ROP RGB new==old} has three possible values.  
\texttt{new < old}, \texttt{new == old}, and \texttt{new > old}.
_tests each of these cases.

Errors in this subtest are attributed to the 3DRAM.
XChip Test

X Chip registers are tested using simple read/write patterns to determine if there are any bad bits. This includes all LUTs. gfbtest ensures that data is actually being read from the RAMDAC and not being supplied by the driver. Next, the RAMDAC Signature Register captures the pixels going to the screen. This test determines that all of the different data paths within the RAMDAC are functioning properly. The following modes are tested:

- 24-bit true color from A
- 24-bit true linear color from A
- 24-bit direct color from A
- 24-bit true color from B
- 24-bit true linear color from B
- 24-bit direct color from B
- 8-bit pseudo color (from each plane in RGB) from A
- 8-bit pseudo color (from each plane in RGB) from B
- 8-bit non-linear grayscale (from each plane in RGB) from A
- 8-bit non-linear grayscale (from each plane in RGB) from B
- 8-bit linear grayscale (from each plane in XRGB) from A
- 8-bit linear grayscale (from each plane in XRGB) from B
- 8-bit overlay pseudo color (from buffer A, X plane)

Errors in this test are attributed to the RAMDAC.

Cafe Test

This test will do non-destructive testing of cafe memory (RDRAM) and cafe. The errors in this test are attributed to the cafe and its memory.

Texture Memory Test

Texture memory test tests out all of the texture memory by writing the data pattern selected (random, 0s, 1s, 5s or 0xAs). By default Random data is selected. The data is written using block writes and read back using block reads. This test automatically detects if the board has 64 MB or 256 MB of texture memory and tests it accordingly.

Errors in this test are attributed to texture memory and texture memory subsystem.
Chapter 27 Graphics Frame Buffer Test (gfbtest)

Rendering Pipeline Test
Each primitive is tested thoroughly by exercising the following:
- Simple Triangles
- 2d primitives
- 3d Primitives (like Triangles, 3d lines etc.)
- Vertex Processor

Errors in this test are attributed to the FBC3.

Texture Pipeline Test
This test renders textured primitives to test
- 2d texture Minification filtering
- 2d texture Magnification filtering
- 3d texture Minification filtering
- 3d texture Magnification filtering
- texture environment
- Filter4 and sharpen filters
- anisotropic filter

Errors in this test are attributed to FBC3.

Fragment Processor Test
Fragment Processor, a subtest, exercises the following options selected by the GFB’s Fragment Processor Control (FPC) register:
- Auxiliary clipping (additive and subtractive)
- Depth cueing
- Alpha blend
- Viewport clip (2D and 3D)
- Area pattern (transparent and opaque)

Errors in this test are attributed to the FBC3.

Lighting Test
The Lighting test exercises GFB float and lighting microcode. This test lights an object with maximum number of lights (32) that GFB can handle in hardware. A checksum is generated for the rendered image and compared with the checksum generated for the same image on a known good system.

Errors in this test are attributed to the Cafe, Microcode and RD RAMs.

Super Sampling Test
This test will test the super sampling filtering. A picture is drawn into off screen memory, then it is filtered through the super sample filter and copied into on screen video memory.

The errors in this test are attributed to FBC3 and 3DRAMs.

TABLE 27-1 gfbtest Options

<table>
<thead>
<tr>
<th>gfbtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rendering Pipeline Test</td>
<td>Each primitive is tested thoroughly by exercising the following:</td>
</tr>
<tr>
<td>Texture Pipeline Test</td>
<td>This test renders textured primitives to test</td>
</tr>
<tr>
<td>Fragment Processor Test</td>
<td>Fragment Processor, a subtest, exercises the following options selected by the GFB’s Fragment Processor Control (FPC) register:</td>
</tr>
<tr>
<td>Lighting Test</td>
<td>The Lighting test exercises GFB float and lighting microcode. This test lights an object with maximum number of lights (32) that GFB can handle in hardware. A checksum is generated for the rendered image and compared with the checksum generated for the same image on a known good system.</td>
</tr>
<tr>
<td>Super Sampling Test</td>
<td>This test will test the super sampling filtering. A picture is drawn into off screen memory, then it is filtered through the super sample filter and copied into on screen video memory.</td>
</tr>
</tbody>
</table>
Mesh Buffer Test

This test exercises the mesh buffer by setting up the mesh buffer and rendering triangles by using mesh buffer.

Errors in this test are attributed to mesh buffer.

Clip Trap Test

Clip trap feature is tested by drawing triangles which cross the clip region. When the triangle falls beyond clip region, a clip trap is sent to cafe. Then, Cafe services the clip trap.

The errors in this test are attributed to cafe and microcode.

Context Switching Test

This test tests the microcode context switching capabilities.

Errors in this test are attributed to Cafe, RDRAMs and/or Microcode.

Mixed Primitives Test

The GFB Mix test draws different primitives with variety combinations of sources and configurations, exercising all the FBC3, Cafe, Microcode, SDRAM and 3DRAM chips on GFB. This test is to stress the GFB.

Errors in this test are attributed to FBC3, CAFE, Microcode, SDRAM, RDRAM and/or 3DRAM Chips.

Picking Test

The Picking test exercises the pick detect login of the 3DRAM. A pick detect window is defined and the test verifies that writes to the window are picked, and writes outside the window are not picked. The test is repeated once for each 3DRAM.

Errors in this test are attributed to the 3DRAM.

Stereo Test

Stereo test displays an object in stereo mode with different images for the right and left eye. You can verify proper operation by looking at the screen with stereo glasses and following the instructions displayed. If the monitor type is not 1280x1024 at 76MHz, this test prints a warning message and does not execute.

To prevent this message from being displayed or written to the SunVTS information log, disable the stereo test in the Test Parameter Options dialog box. This test temporarily switches the monitor into stereo mode, renders a stereo image, performs a signature analysis on the stereo image (using the RAMDAC signature capture register), and after displaying the image for five seconds, restores the monitor to its previous resolution.

Errors in this test are attributed to the X Chip.

<table>
<thead>
<tr>
<th>Table 27-1</th>
<th>gfbtest Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>gfbtest Options</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Mesh Buffer Test</td>
<td>This test exercises the mesh buffer by setting up the mesh buffer and rendering triangles by using mesh buffer.</td>
</tr>
<tr>
<td>Clip Trap Test</td>
<td>Clip trap feature is tested by drawing triangles which cross the clip region. When the triangle falls beyond clip region, a clip trap is sent to cafe. Then, Cafe services the clip trap.</td>
</tr>
<tr>
<td>Context Switching Test</td>
<td>This test tests the microcode context switching capabilities.</td>
</tr>
<tr>
<td>Mixed Primitives Test</td>
<td>The GFB Mix test draws different primitives with variety combinations of sources and configurations, exercising all the FBC3, Cafe, Microcode, SDRAM and 3DRAM chips on GFB. This test is to stress the GFB.</td>
</tr>
<tr>
<td>Picking Test</td>
<td>The Picking test exercises the pick detect login of the 3DRAM. A pick detect window is defined and the test verifies that writes to the window are picked, and writes outside the window are not picked. The test is repeated once for each 3DRAM.</td>
</tr>
<tr>
<td>Stereo Test</td>
<td>Stereo test displays an object in stereo mode with different images for the right and left eye. You can verify proper operation by looking at the screen with stereo glasses and following the instructions displayed. If the monitor type is not 1280x1024 at 76MHz, this test prints a warning message and does not execute.</td>
</tr>
</tbody>
</table>

To prevent this message from being displayed or written to the SunVTS information log, disable the stereo test in the Test Parameter Options dialog box. This test temporarily switches the monitor into stereo mode, renders a stereo image, performs a signature analysis on the stereo image (using the RAMDAC signature capture register), and after displaying the image for five seconds, restores the monitor to its previous resolution. |

Errors in this test are attributed to the X Chip.
**gfbtest Test Modes**

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

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests</td>
</tr>
</tbody>
</table>

**gfbtest Command Line Syntax**

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>device_name is the relative path name of the device being tested with respect to /dev/fbs. The default is gfb0.</td>
</tr>
</tbody>
</table>
### TABLE 27-3  
**gfbtest Command-Line Syntax**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| $s$=subtest_number       | subtest_number is the test number of the subtest to be run. Select from the subtests below. You can run multiple subtests by adding the subtest numbers together. For example, $n=0x3$ runs both test 1 and test 2; $n=0x180$ runs both test 0x080 and test 0x0100. You do not need the leading zeros.  
  * $n$—0x00001 Video Memory 3DRAM  
  * $n$—0x00002 3DRAM Logic  
  * $n$—0x00004 X Chip  
  * $n$—0x00008 Cafe  
  * $n$—0x00010 Texture Memory SDRAM  
  * $n$—0x00020 Rendering Pipeline  
  * $n$—0x00040 Texturing Pipeline  
  * $n$—0x00080 Fragment Processor  
  * $n$—0x00100 Lighting  
  * $n$—0x00200 Super Sampling  
  * $n$—0x00400 Mesh Buffer  
  * $n$—0x00800 Clip Trap  
  * $n$—0x01000 Context Switching  
  * $n$—0x02000 Mixed Primitives  
  * $n$—0x04000 Picking  
  * $n$—0x08000 Stereo |
| F=#_of_subtest_loops     | The number of times to repeat each subtest.  
The default is 1. |
| B=#_of_test_loops       | The number of times to repeat a test loop before passing.  
The default is 1. |
| P=test_pattern          | The test pattern number. The default is $r$, for random patterns.  
You may also choose 0 for 0x0000000, 3 for 0x3333333, 5 for 0x5555555, or 9 for 0x9999999. |

**Note** — If looping on a test, the verbose mode is disabled. Separate multiple parameters with commas. Example: gfbtest -o dev=gfb1,$s=0x9,B=2

**Note** — 64-bit tests are located in the sparcv9 subdirectory: /opt/SUNWvts/bin/sparcv9/testname. If the test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to “32-Bit and 64-Bit Tests” on page 13.
Note — Errors returned by `gfbtest` are nonspecific: It is not possible to determine which component caused a failure. In all error conditions, the field replaceable unit (FRU) is the entire GFB.
The gfxtest tests the PGX32 frame buffer card by performing video memory, RAMDAC, and acceleration subtests as described in TABLE 28-1.

TABLE 28-1  gfxtest Subtests

<table>
<thead>
<tr>
<th>Subtest</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Memory</td>
<td>Thoroughly tests the 8-MB of video frame buffer memory using random and sequential 8-bit and 32-bit accesses. One pass of this subtest takes a while to complete.</td>
</tr>
<tr>
<td>RAMDAC</td>
<td>Tests the internal circuitry of the video output. The RAMDAC CLUT (color Lookup table) is tested using simple/read-write patterns to determine if there are any bad bits in the CLUT. This test only checks the functionality of the digital portion of the RAMDAC. This subtest does not ensure that the analog video signals are produced properly. This subtest takes less then one second to complete.</td>
</tr>
<tr>
<td>Acceleration</td>
<td>Thoroughly tests the hardware graphics acceleration of the PGX32 card. It re-initializes the engine, tests primitives such as lines, rectangles, bitblts, tiled and stippled rectangles with on-screen and off-screen memory. The Acceleration test also tests the ALU. This test takes less than one second to complete.</td>
</tr>
</tbody>
</table>

Caution – DO NOT run any other application or screen saver program that uses the PGX32 frame buffer card. These programs cause SunVTS to return incorrect errors.
Disable all screen savers before testing any graphics device. Type `xset s off` at a UNIX prompt to disable the Solaris screen saver.

Due to the nature of graphics tests, reading from or writing to the frame buffer during graphics tests will disturb user operation.

Do not run `gfxtest` from the SunVTS TTY mode when you are at the console.

For full instructions on testing frame buffers, see “Testing Frame Buffers” on page 18.

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

---

**gfxtest Options**

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User’s Guide* for more details.
FIGURE 28-1  gfxtest Test Parameter Options Dialog Box

TABLE 28-2  gfxtest Options

<table>
<thead>
<tr>
<th>Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Test</td>
<td>Enables or disables the Video Memory subtest. The default is enabled.</td>
</tr>
<tr>
<td>RAMDAC Test</td>
<td>Enables or disables the RAMDAC subtest. The default is enabled.</td>
</tr>
<tr>
<td>Acceleration Test</td>
<td>Enables or disables the Acceleration subtest. The default is enabled.</td>
</tr>
</tbody>
</table>
gfxtest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full set of tests.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

gfxtest Command-Line Syntax

```
/opt/SUNWvts/bin/gfxtest  standard_arguments  -o  dev=device_name,mem=
no_of_passes,ramdac=no_of_passes,accel=no_of_passes
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td><code>device_name</code> is the relative path name of the device being tested with respect to /dev/fbs, for example, /dev/fbs/gfxp0.</td>
</tr>
<tr>
<td>mem=no_of_passes</td>
<td>Specifies the number of times to run the Video Memory subtest. The default is one time.</td>
</tr>
<tr>
<td>ramdac=no_of_passes</td>
<td>Specifies the number of times to run the RAMDAC subtest. The default is one time.</td>
</tr>
<tr>
<td>accel=no_of_passes</td>
<td>Specifies the number of times to run the Acceleration subtest. The default is one time.</td>
</tr>
</tbody>
</table>

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 “32-Bit and 64-Bit Tests” on page 13.

Note – Errors returned by gfxtest are nonspecific. It is not possible to determine which component caused a failure. In all error conditions, the field replaceable unit (FRU) is the entire PGX32 frame buffer card.
I2C Bus Test (i2ctest)

i2ctest checks each I2C bus for the status of its devices. i2ctest then determines any device faults based on the information it collects, and displays a report. i2ctest also detects and reports hung I2C bus segments.

This test is scalable.

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

### i2ctest Test Requirements

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

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

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

```
# /etc/init.d/picld start
```
i2ctest Options

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

FIGURE 29-1 i2ctest Test Parameter Options Dialog Box
**i2ctest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Runs the full test.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full test.</td>
</tr>
</tbody>
</table>

**i2ctest Command-Line Syntax**

```
/opt/SUNWvts/bin/sparcv9/i2ctest standard_arguments
-o dev=raw_device_name
```

**TABLE 29-2** i2ctest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=raw_device_name</td>
<td>Specifies the name of the raw device to test.</td>
</tr>
</tbody>
</table>
**Expert3D Frame Buffer Test (ifbtest)**

`ifbtest` verifies the functionality of the Expert3D frame buffer.

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

You can interrupt `ifbtest` using Control-C.

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

---

**Caution** – Do not run any other application or screen saver program that uses the Expert3D accelerator port while running `ifbtest`. This combination causes SunVTS to return incorrect errors.

---

**ifbtest Test Requirements**

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

```bash
# xset s off
```

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

```
# xset -dpms
```

For full instructions on testing frame buffers, see “Testing Frame Buffers” on page 18.

**Preparation for ifbtest**

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

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

- Turn Power Management off, if it is enabled. The following is an alternate way to turn Power Management off. Change allowFBPM=1 to allowFBPM=0 in /platform/sun4u/kernal/drv/ifb.conf file.
- Make sure that no other program is running that might modify the screen during the test.
- Make sure you have permission to lock the X server. ifbtest is designed to lock the X server during testing to prevent screen changes.
- The CDE login window should not be displayed during testing.
- Check that the window system is only running on one Expert3D frame buffer.

If you are not running ifbtest in a window system:

- Turn Power Management off, if it is enabled. The following is an alternate way to turn Power Management off. Change allowFBPM=1 to allowFBPM=0 in /platform/sun4u/kernal/drv/ifb.conf file.
- Make sure that no other program is running that might modify the screen during the test.
- Make sure the Expert3D frame buffer being tested is not the console device. Console messages may modify the screen.
ifbtest Options

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

By default, all ifbtest options are enabled.

![ifb0(ifbtest) Test Parameter Options Dialog Box](image)

**FIGURE 30-1** ifbtest Test Parameter Options Dialog Box
<table>
<thead>
<tr>
<th>ifbtest Options</th>
<th>Description</th>
</tr>
</thead>
</table>
| Frame Buffer Memory test| Thoroughly tests the Expert3D video memory by using read and write requests. Tests for shorts or failed connections on the data bus by writing the following values to every address:  
  - 0xFFFFFFFF  
  - 0xFFFF0000  
  - 0x0000FFFF  
  - 0xFFF0FF00  
  - 0x00FF00FF  
  - 0xFF00FF00  
  - 0x0F0F0F0F  
  - 0xCCCCCCCC  
  - 0x33333333  
  - 0xAяАяяйяя  
  - 0x55555555  
  Tests for shorts or failed connections on the address bus by writing the offset of each memory location to each location and reading them back. This may also catch speed-related problems due to the volume of read/writes.  
  Errors in the test are reported as an error in a particular address, not attributed to a specific chip. To help distinguish bit-related errors, the errors are summarized to list which bits had at least one error in the test.  
  This test shows on the screen as random pixels. |
| Texture Memory test      | This test is identical in process to the frame buffer memory test (above). Since this test produces no visible effect, rectangles are drawn in rows across the screen to show progress. |
| Display List Memory test | This test is identical in process to the frame buffer memory and texture memory tests (above), and is applied to direct burst memory.  
  This test takes little time and no progress is displayed. |
| Geometry Engine test     | Loads diagnostic microcode into the geometry engine and confirms that the processor operates correctly. This is a pass/fail test.  
  This test takes little time and no progress is displayed. |
Chapter 30  Expert3D Frame Buffer Test (ifbtest)

Rasterization test

Renders many primitives with minimal fragment processing, to test the rasterization of the primitives.

The primitives used are:
• Dots
• Anti-aliased dots
• Lines using all for line-drawing primitives
• Anti-aliased lines using all for line-drawing primitives
• Triangles, Quads, and Polygons in point, line, and fill modes
• Rectangles

This tests for the following rasterization attributes:
• pixel coverage
• constant value registers for color, Z, and stencil
• interpolation of color, Z, and texture coordinates along lines and spans in polygons
• texture map sampling

Resulting images are compared against stored images. Errors indicate which operation type and value was being tested, and the coordinate of the failed pixel.

<table>
<thead>
<tr>
<th>ifbtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rasterization test</td>
<td>Renders many primitives with minimal fragment processing, to test the rasterization of the primitives.</td>
</tr>
</tbody>
</table>
Pixel Processor test

Tries the various pixel processing operators using a variety of fragment values. This tests the following fragment processing operations:

- Depth Buffering
- Blending
- Alpha Test
- Color Test
- Color Clamp
- Logic Operations
- Color Matrix and Bias
- Color Table
- Control Planes
- Fast Clear
- Stencil
- Scissor Clipping
- Desktop Clipping
- Mask Clipping
- Write Masks
- Window Origin
- Fog
- Pixel Texture
- Accumulation Buffer
- Pixel Buffers

Resulting images are compared against stored images. Errors indicate which operation type and value was being tested and the coordinate of the failed pixel.

<table>
<thead>
<tr>
<th>ifbtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel Processor test</td>
<td>Tries the various pixel processing operators using a variety of fragment values. This tests the following fragment processing operations:</td>
</tr>
<tr>
<td></td>
<td>Depth Buffering</td>
</tr>
<tr>
<td></td>
<td>Blending</td>
</tr>
<tr>
<td></td>
<td>Alpha Test</td>
</tr>
<tr>
<td></td>
<td>Color Test</td>
</tr>
<tr>
<td></td>
<td>Color Clamp</td>
</tr>
<tr>
<td></td>
<td>Logic Operations</td>
</tr>
<tr>
<td></td>
<td>Color Matrix and Bias</td>
</tr>
<tr>
<td></td>
<td>Color Table</td>
</tr>
<tr>
<td></td>
<td>Control Planes</td>
</tr>
<tr>
<td></td>
<td>Fast Clear</td>
</tr>
<tr>
<td></td>
<td>Stencil</td>
</tr>
<tr>
<td></td>
<td>Scissor Clipping</td>
</tr>
<tr>
<td></td>
<td>Desktop Clipping</td>
</tr>
<tr>
<td></td>
<td>Mask Clipping</td>
</tr>
<tr>
<td></td>
<td>Write Masks</td>
</tr>
<tr>
<td></td>
<td>Window Origin</td>
</tr>
<tr>
<td></td>
<td>Fog</td>
</tr>
<tr>
<td></td>
<td>Pixel Texture</td>
</tr>
<tr>
<td></td>
<td>Accumulation Buffer</td>
</tr>
<tr>
<td></td>
<td>Pixel Buffers</td>
</tr>
</tbody>
</table>
ifbtest Test Modes

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

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full set of tests.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ifbtest Command-Line Syntax

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

TABLE 30-3 ifbtest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>device_name is the relative path name of the device being tested with respect to /dev/fbs. There is no default.</td>
</tr>
<tr>
<td>fbmem=E/D</td>
<td>Enables or disables the frame buffer memory test.</td>
</tr>
<tr>
<td>texmem=E/D</td>
<td>Enables or disables the texture memory test.</td>
</tr>
<tr>
<td>dimem=E/D</td>
<td>Enables or disables the display list memory test.</td>
</tr>
<tr>
<td>geomeng=E/D</td>
<td>Enables or disables the geometry engine test.</td>
</tr>
<tr>
<td>rasterization=E/D</td>
<td>Enables or disables the rasterization test.</td>
</tr>
<tr>
<td>pixelproc=E/D</td>
<td>Enables or disables the pixel processing test.</td>
</tr>
<tr>
<td>subtest_repeat=number</td>
<td>Defines the number of times to repeat each subtest. The default is 1.</td>
</tr>
<tr>
<td>test_repeat=number</td>
<td>Defines the number of times to repeat a test loop before passing. The default is 1.</td>
</tr>
</tbody>
</table>
Note – 64-bit tests are located in the sparcv9 subdirectory: `/opt/SUNWvts/bin/sparcv9/testname`. If the test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to “32-Bit and 64-Bit Tests” on page 13.
Intelligent Fibre Channel Processor Test (ifptest)

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

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

ifptest uses the “mailbox” interface to the card. This interface allows certain firmware operations to be performed that normally would not be available to the application layer.

**ifptest Subtests**

Four subtests are run in online and functional modes:

- Mailbox Loopback test

  Loads a series of registers into the input mailboxes on the card and then reads the output mailboxes and compares results. This verifies that the system side of the card is operating correctly, and that the internal data paths are okay.

- Firmware revision check

  Reads the firmware revision from the firmware and compares it against a revision loaded by the driver.
- Checksum firmware test

Runs an internal checksum test on the installed firmware. This verifies that the RISC RAM on the card is fully functional and that the installed firmware is still intact. This test also serves as a quick RAM check of the RISC RAM.

- Dump revision levels

Extracts the hardware and firmware revision levels of different submodules on the card.

---

**ifptest Options**

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User’s Guide* for more details.
FIGURE 31-1  ifptest Test Parameter Options Dialog Box

TABLE 31-1  ifptest Options

<table>
<thead>
<tr>
<th>ifptest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailbox Loopback test</td>
<td>Enables or disables the mailbox loopback command. This test writes data patterns into the mailboxes and then reads them back from the output mailboxes and verifies the data is correct.</td>
</tr>
</tbody>
</table>
ifptest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Performs only an open/close operation.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of mailbox tests.</td>
</tr>
</tbody>
</table>

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

---

**TABLE 31-1 ifptest Options**

<table>
<thead>
<tr>
<th>ifptest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware revision check</td>
<td>Enables or disables the firmware revision check command. This test extracts the firmware revision from the RISC firmware code and verifies against expected values.</td>
</tr>
<tr>
<td>Firmware checksum test</td>
<td>Enables or disables the firmware checksum command. This command instructs the interface's RISC processor to calculate the checksum on the current microcode and then compare it against the checksum that was loaded in with the microcode.</td>
</tr>
<tr>
<td>Check Module Revisions</td>
<td>Enables or disables the firmware check module command. This command returns the revision level of several submodules on the interface card. Although this test is executed when enabled, the module revision levels are only printed out in VERBOSE mode.</td>
</tr>
</tbody>
</table>

**TABLE 31-2 ifptest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Performs only an open/close operation.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full set of mailbox tests.</td>
</tr>
</tbody>
</table>

**Note** – Connection test mode will only open the controller to verify that the path is still viable.
ifptest Command-Line Syntax

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

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

**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 “32-Bit and 64-Bit Tests” on page 13.
Level 1 Data Cache Test (l1dcachetest)

l1dcachetest exercises the level1 D cache in the CPU module of a Sun system. The test writes, reads, and verifies access of multiple virtual addresses. The test determines addresses by the buffer size, which is chosen according to the size of the l1dcache. The test writes to an address bigger than the cache size, to cause thrashing (cache misses) in up to 4-way set associative data caches.

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.
**TABLE 32-1  l1dcachetest Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thrash Cycles</td>
<td>Specifies the number of thrashing cycles the test completes for the level1 cache on the system. Default value for Offline mode is 256.</td>
</tr>
<tr>
<td>Test Buffer Size (in kbytes)</td>
<td>Sets the size of the buffer, in kbytes, that the test allocates for testing. Default value is 64.</td>
</tr>
</tbody>
</table>
l1dcachetest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full test.</td>
</tr>
</tbody>
</table>

l1dcachetest Command-Line Syntax

```
/opt/SUNWvts/bin/l1dcache standard_arguments
-o count=number, buffer=number, dev=l1dcache
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>count=number</td>
<td>Specifies the number of thrashing cycles the test completes for the level1 cache on the system. Default value for Offline mode is 256.</td>
</tr>
<tr>
<td>buffer=number</td>
<td>Sets the size of the buffer, in kbytes, that the test allocates for testing. Default value is 64.</td>
</tr>
<tr>
<td>dev=l1dcache</td>
<td>Specifies the name of the device.</td>
</tr>
</tbody>
</table>
Level 2 Cache Test
(l2dcachetest)

l2dcachetest exercises the level2 external cache in the CPU module of a Sun machine. The test writes, reads, and verifies access of multiple virtual addresses. The multiple virtual addresses are chosen to cause thrashing (cache misses) in direct-mapped external caches.

l2dcachetest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
**FIGURE 33-1 l2dcachetest Test Parameter Options Dialog Box**

**TABLE 33-1 l2dcachetest Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thrash Cycles</td>
<td>Specifies the number of thrashing cycles the test completes for the level2</td>
</tr>
<tr>
<td></td>
<td>cache on the system. Default value for Offline mode is 8.</td>
</tr>
<tr>
<td>Test Buffer Size</td>
<td>Sets the size of the buffer, in kbytes, that the test allocates for testing.</td>
</tr>
<tr>
<td></td>
<td>Default value is 8192.</td>
</tr>
</tbody>
</table>
12dcachetest Test Modes

TABLE 33-2 12dcachetest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full test. (Offline)</td>
</tr>
</tbody>
</table>

12dcachetest Command-Line Syntax

```
/opt/SUNWvts/bin/12dcache standard_arguments
-o count=number, buffer=number, dev=12dcache
```

TABLE 33-3 12dcachetest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>count=number</code></td>
<td>Specifies the number of thrashing cycles the test completes for the level2 cache on the system. Default value for Offline mode is 8.</td>
</tr>
<tr>
<td><code>buffer=number</code></td>
<td>Sets the size of the buffer, in kbytes, that the test allocates for testing. Default value is 8192.</td>
</tr>
<tr>
<td><code>dev=12dcache</code></td>
<td>Specifies the name of the device.</td>
</tr>
</tbody>
</table>
LOMlite Alarm Test
(lomlitetest)

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

This test is not scalable.

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

lomlitetest Requirements

- The LOMlite or TSalarms device driver must be installed
- In the Netra t 11xx or Netra t 14xx platforms, the LOMlite or TSalarms plug-in card must be installed
### lomlitetest Subtests

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

TABLE 34-1  lomlitetest Subtests
lomlitetest Options

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

FIGURE 34-1 shows the options menu for a multi-processor system. 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.

![lomlite(lomlitetest) Test Parameter Options Dialog Box, Multi-processor System](image)
**lomlitetest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Runs the Connection subtest.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs all subtests.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**lomlitetest Command-Line Syntax**

```
/opt/SUNWvts/bin/lomlitetest standard_arguments
-o dev=lomlite2 | lomlite | tsalarms
```

**TABLE 34-3 lomlitetest Command-Line Syntax**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=</td>
<td>Selects the type of device driver to test.</td>
</tr>
</tbody>
</table>
| lomlite2 | lomlite | tsalarms
```
M64 Video Board Test (m64test)

m64test tests the PCI-based M64 video board by performing the following subtests:

- Video Memory test
- RAMDAC test
- Accelerator Port test

**Caution** – Do not run any other application or screen saver program that uses the M64 video board while running m64test. Do not run Power Management™ software. These programs cause SunVTS to return incorrect errors.

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

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

For full instructions on testing frame buffers, see “Testing Frame Buffers” on page 18.
m64test Options

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

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

![m64test Test Parameter Options Dialog Box](image)

**FIGURE 35-1 m64test Test Parameter Options Dialog Box**
### TABLE 35-1  m64test Options

<table>
<thead>
<tr>
<th>m64test Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Memory test</td>
<td>Thoroughly tests the on-screen video memory (the memory part that is mapped on to the monitor) of the M64 video board in 8-bit, 16-bit, 32-bit, 64-bit, and 64 byte (block) modes. Entire on-screen video memory is tested by testing 512 bit blocks at a time (8x8 pixel block). Each block is tested in two passes. Each pass consists of a data write and read. In the first pass user specified data or random data is used and in the second pass one’s complement of the data used in the first pass is used so that each on-screen video memory location (bit) is tested with a zero (electrical low state) and one (electrical high state).</td>
</tr>
</tbody>
</table>
| RAMDAC test | Tests the RAMDAC in three phases. In the first phase the RAMDAC CLUT (Color LookUp Table) is tested using simple write/read patterns to determine if there are any bad bits in CLUT. The data patterns used are:  
  - Random data  
  - Complement of the random data (used as first data pattern)  
  - The data pattern 0101  
  - The data pattern 10101  
  
  In the second phase, four different patterns are drawn on the screen. Each pattern stays on the screen for approximately three seconds. The four patterns are listed below. For each pattern the signature is captured and compared with the signature obtained for the same pattern on a known good board. This test verifies that all the different data paths within the RAMDAC are functioning properly.  
  
  Patterns drawn on screen:  
  - Red ramp with cursor at top-left corner of the screen  
  - Blue ramp with cursor at top-right corner of the screen  
  - Green ramp with cursor at bottom-left of the screen  
  - Grey ramp with cursor at bottom-right of the screen  
  
  In the last (third) phase of the RAMDAC test the Vertical Retrace Interrupt is tested for 300 interrupts. |
m64test Test Modes

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

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>The m64test verifies the M64 video board.</td>
</tr>
</tbody>
</table>
m64test Command-Line Syntax

```
/opt/SUNWvts/bin/m64test standard_arguments -o dev=device_name, S=subtest_number,F=#_of_subtest_loops,B=#_of_test_loops,L=disable,P=test_pattern
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>device_name is the relative path name of the device being tested with respect to /dev/fbs. The default is m640.</td>
</tr>
</tbody>
</table>
| S=subtest_number | subtest_number is the test number of the subtest to be run. Select from the subtests below. You can run multiple subtests by adding the subtest numbers. For example, n=0x00003 runs both test 00001 and test 00002; n=0x00005 runs both test 0x00001 and test 0x00004. Note that you do not need the leading zeros.  
  • n=0x00001 VRAM  
  • n=0x00002 RAMDAC  
  • n=0x00004 Accelerator port test (Rendering Pipeline)  
  More than one test can be selected by ORing subtest numbers. For example: n = 0x00005 means VRAM and Rendering Pipeline tests. A hex number must be preceded by 0x, decimal numbers are also acceptable. |
| F=#_of_subtest_loops | Specifies the number of times to repeat each subtest. The default is 1. |
| B=#_of_test_loops | Specifies the number of times to repeat a test loop before passing; default is 1. |
| L=disable | Disables the frame buffer lock. Disable the lock when the m64 is not the console or when the server is not running on the m64 under test. |
| P=test_pattern | Specifies the test pattern number. The default is r, for random patterns. You may also choose 0 for 0x0000000, 3 for 0x3333333, 5 for 0x5555555, or 9 for 0x9999999. |

Note – 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 “32-Bit and 64-Bit Tests” on page 13.
Note – Errors returned by m64test are nonspecific. It is not possible to determine which component caused a failure. In all error conditions, the field replaceable unit (FRU) is the entire M64 video board.
CHAPTER 36

Cache Consistency Test (mpconstest)

mpconstest verifies that cache coherency is maintained in a multi-processor environment by creating contention for one or more cache lines.

Only the following cpus are supported:
- SuperSPARC [TI] (SS10/SS20/SS1000/SC2000)
- SuperSPARC II
- MicroSPARC II [TI] (50 MHz)
- MicroSPARC II (SS5)
- UltraSPARC I [TI] (143/167/200MHz)
- UltraSPARC II (250/333/336/360MHz)
- UltraSPARC III (500-600/750/900MHz)
- UltraSPARC III Cu

This test has several subtests, each designed to create a different kind of contention for cache lines. Each subtest uses different methods to test the shared memory buffer, the stride size, and any intermediate stores or loads.

When mpconstest starts, it creates a shared memory buffer. It then determines the number of CPUs on the system. For each CPU, the test takes the following steps:

1. Forks a thread and binds it to the CPU.
2. Runs the selected subtest in the thread.
3. Assigns each CPU an ID number from 1 to n. The CPU assigned ID 1 is considered the master.

The above steps are repeated for each subtest. Only one subtest can be selected at a time.

This test is not scalable.
**mpconstest Test Requirements**

This test requires that the tested system has at least two CPUs. Otherwise, the test will not appear as an option.

`mpconstest` only runs on machines that support the v8plus standard of SPARC CPU hardware architecture. If the v8plus instructions are not supported, `mpconstest` will not appear on the Test Selection GUI. To determine whether a machine supports the v8plus standard, go to a command prompt on that machine and type:

```
isalist
sparcv9+vis sparcv9 sparcv8plus+vis sparcv8plus sparcv8
```

**Note** – This set of tests is very sensitive to activity on the machine and must be run exclusive of all other tests.
### mpconstest Subtests

<table>
<thead>
<tr>
<th>Table 36-1  mpconstest subtests</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tests</strong></td>
</tr>
<tr>
<td>cons1</td>
</tr>
<tr>
<td>cons2</td>
</tr>
<tr>
<td>cons3</td>
</tr>
<tr>
<td>cons4</td>
</tr>
<tr>
<td>cons5</td>
</tr>
<tr>
<td>cons6</td>
</tr>
<tr>
<td>cons7</td>
</tr>
</tbody>
</table>
mpconstest 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.

---

<table>
<thead>
<tr>
<th>Tests</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cons8</td>
<td>Similar to cons3 except that each CPU performs one storeb and one loadb operation between the detection of the CPU ID and the write of the next CPU ID. The target of the storeb and loadb is one unique byte of a double word of a private (unshared) line whose line number is identical to the line number containing the IDs. The storeb data is unique to each CPU and changes each time the address of the line containing the IDs changes.</td>
</tr>
<tr>
<td>cons9</td>
<td>Similar to cons8 except that the target of the storeb and loadb is one unique byte of a double word of a private line whose address does not change through the entire test.</td>
</tr>
<tr>
<td>cons10</td>
<td>Similar to cons9 except that two storeb and two loadb operations are performed to private (unshared) lines. The target of the second storeb is cachesize bytes away from the target of the first storeb. In a direct map cache, this results in a writeback of the unshared data written with the first storeb. The loadb operations are performed after the storeb in order to ensure that the writeback occurs correctly.</td>
</tr>
<tr>
<td>cons11</td>
<td>Similar to cons10 except that the target of the storeb and loadb operations is to a shared line rather than a private line.</td>
</tr>
<tr>
<td>cons12</td>
<td>Similar to cons7 except that two store double (stored) and load double (loadd) operations are used in place of the storeb and loadb operations. The target of the stored and loadd operations are two consecutive double words of a shared line. This test is designed to verify that the double word operations are performed correctly while the shared and owned state of the line containing the ID is changing.</td>
</tr>
<tr>
<td>cons13 through cons17</td>
<td>These tests are similar variations of intermediate operations, stride size etc, and do not involve any new interfaces.</td>
</tr>
</tbody>
</table>
FIGURE 36-1 mpconstest Test Parameter Options Dialog Box
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Name</td>
<td>Selects the subtest to be run.</td>
</tr>
<tr>
<td>Number of Timeout Iterations</td>
<td>Sets the number of times the test is allowed to time out. Default is 1. Note that each timeout occurs after a greater amount of elapsed time than the previous one. That is, if the first timeout occurs after T units of time, the second occurs 2T after T, and the third occurs 3T after 2T.</td>
</tr>
<tr>
<td>Lock Buffer</td>
<td>Locks Buffer in Memory. Default is not locked. Locking the buffer in memory will disable COMA (Cache Only Memory Architecture).</td>
</tr>
<tr>
<td>Atomic Mode</td>
<td>Uses the atomic instruction swap. Default is disabled.</td>
</tr>
<tr>
<td>Byte Mode</td>
<td>Uses byte instructions to load and store. Default is disabled.</td>
</tr>
<tr>
<td>Immediate Mode</td>
<td>Supports all subtests except cons1, cons2, cons3, cons15, cons16, and cons17.</td>
</tr>
<tr>
<td>Random Mode</td>
<td>Enables Random Mode.</td>
</tr>
<tr>
<td>Reverse Mode</td>
<td>Traverses the shared memory buffer in reverse. Default is disabled.</td>
</tr>
<tr>
<td>Prefetch Mode</td>
<td>Sets prefetch for read and write. Default is disabled.</td>
</tr>
<tr>
<td>CoreFile</td>
<td>Generates a core file. Exits in case of unexpected signals. Default is disabled.</td>
</tr>
<tr>
<td>Ecache Disable</td>
<td>Disables the external cache. Default is enabled.</td>
</tr>
<tr>
<td>Trigger</td>
<td>Sends an interrupt signal to all processors when one processor detects a failure. Default is disabled.</td>
</tr>
<tr>
<td>Offset</td>
<td>Specifies an offset of line size between successive writes. Default is disabled.</td>
</tr>
<tr>
<td>CPU Wait Count</td>
<td>Forces CPU 1 to write first if the number of CPUs is less than cpucount. Default is disabled.</td>
</tr>
<tr>
<td></td>
<td>This option is not supported with subtests cons15, cons16, and cons17.</td>
</tr>
<tr>
<td>Number of Loops</td>
<td>Selects the number of test loops. Default is 5.</td>
</tr>
<tr>
<td>Number of Passes</td>
<td>Selects the number of passes. Increasing the number of passes increases system stress. Setting the number of passes to 0 will cause the test to run in an endless loop. Passes can only be set to 0 in command line mode, not from the GUI. Default is 1.</td>
</tr>
<tr>
<td>Memory Size</td>
<td>Selects the memory size, in Megabytes, for the shared buffer. Default is 128.</td>
</tr>
<tr>
<td>Random Mode Seed</td>
<td>Sets random number seed to a user specified value. Selects a random number seed by default.</td>
</tr>
</tbody>
</table>
**mpconstest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full test.</td>
</tr>
</tbody>
</table>

**mpconstest Command-Line Syntax**

```
/opt/SUNWvts/bin/mpconstest standard_arguments
-o tst=Cons1|Cons2,itm=number,lck,a,b,c,e,h,loops=number,memsizememsizememsize, wait=cpucount,passes=passes,r,t,x,y,i,q,seed=number
```

**TABLE 36-4  mpconstest Command-Line Syntax**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tst</strong>=</td>
<td>Range of choices available between cons1 through cons17 subtests.</td>
</tr>
<tr>
<td>Cons1</td>
<td>Cons2</td>
</tr>
<tr>
<td><strong>itm=number</strong></td>
<td>Sets the number of times the test is allowed to time out. Default is 1. Note that each timeout occurs after a greater amount of elapsed time than the previous one. That is, if the first timeout occurs after ( T ) units of time, the second occurs ( 2T ) after ( T ), and the third occurs ( 3T ) after ( 2T ).</td>
</tr>
<tr>
<td><strong>lck</strong></td>
<td>Locks Buffer in Memory. Default is not locked. Locking the buffer in memory will disable COMA (Cache Only Memory Architecture).</td>
</tr>
<tr>
<td><strong>a</strong></td>
<td>Enables atomic mode. Uses the atomic instruction swap</td>
</tr>
<tr>
<td><strong>b</strong></td>
<td>Enables byte mode. Uses byte instructions to load and store.</td>
</tr>
<tr>
<td><strong>c</strong></td>
<td>Generates a core file. Exits in case of unexpected signals.</td>
</tr>
</tbody>
</table>
TABLE 36-4  mpconstest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>Disables the external cache.</td>
</tr>
<tr>
<td>h</td>
<td>Prints usage message.</td>
</tr>
<tr>
<td>loops=number</td>
<td>Sets the number of loops for the iterations. Default is 5.</td>
</tr>
<tr>
<td>memsize=memb</td>
<td>Selects the memory size, in Megabytes, for the shared buffer. Default is 128.</td>
</tr>
<tr>
<td>wait=cpucount</td>
<td>Forces CPU 1 to write first if the number of CPUs is less than cpucount.</td>
</tr>
<tr>
<td>passes=passe</td>
<td>Selects the number of passes. Increasing the number of passes increases system stress. Setting the number of passes to 0 will cause the test to run in an endless loop. Passes can only be set to 0 in command line mode, not from the GUI. Default is 1.</td>
</tr>
<tr>
<td>r</td>
<td>Enables Reverse mode. Traverses the shared memory buffer in reverse.</td>
</tr>
<tr>
<td>t</td>
<td>Enables Trigger. Sends an interrupt signal to all processors when one processor detects a failure.</td>
</tr>
<tr>
<td>x</td>
<td>Enables Prefetch. Sets prefetch for read and write.</td>
</tr>
<tr>
<td>y</td>
<td>Enables Offset. Specifies an offset of line size between successive writes.</td>
</tr>
<tr>
<td>i</td>
<td>Enables Immediate Mode. Not supported for subtests cons1, cons2, cons3, cons15, and cons 17.</td>
</tr>
<tr>
<td>q</td>
<td>Enables Random Mode.</td>
</tr>
<tr>
<td>seed</td>
<td>Sets a random number seed to the user specified value.</td>
</tr>
</tbody>
</table>
Multiprocessor Test (mptest)

mptest verifies the functionality of multiprocessing hardware. This test allocates a page of virtual memory for the test—declaring the page shared—locks the page against swapping, and creates threads to each of the processors being tested. Up to 1024 processors can be tested by mptest in a CPU.

The processor mask argument can be used during test probing. The mptest verifies that the current processor mask matches the argument you entered in the command line or from the GUI/TTYUI.

mptest 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.
The processors that can be tested are listed in the Configuration area of the menu. You can enable or disable the multiprocessing test for individual processors on this menu.
The options listed in TABLE 37-1 can be run alone or concurrently with other options.

**TABLE 37-1  **mptest **Options**

<table>
<thead>
<tr>
<th><strong>mptest Options</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| **Processors**     | You can test specific processors by clicking Select on the check boxes to enable or disable each processor. A check mark means the processor is enabled for testing. The default setting is all processors enabled.  

Note: mptest requires at least two enabled processors to test multiprocessing systems. |
| **Lock/Unlock**    | Tests the lock/unlock mechanism that guarantees exclusive access to a physical page to one processor. A thread is created at each of the processors. Each processor uses the SPARC atomic instruction **ldstub** to write to the same shared physical memory page. While one processor is attempting the write, the other processors should be free spinning for their turn. As each processor acquires the lock, it writes an ordinal number to a shared trace buffer using a shared write pointer. After the test cycle is complete, the trace buffer is dumped for analysis.  

This test fails and returns an error message if the trace buffer does not contain an equal number of ordinal numbers for each processor. For example, if the specified loop count is 5, the trace buffer should contain five 0s, five negative 1s, five 2s, and so on. |
Data I/O Requires two or more threads, each of which locks onto one of the processors. Each processor, in turn, writes data to a temporary file that has been mapped to the physical address. The modified data is immediately read by other processors being tested. This test hangs and fails if the processors do not recognize the expected data.

Shared Memory A shared memory buffer is divided into a number of contiguous chunks, one for each of the CPUs participating in the test. Each CPU is assigned a unique chunk based upon its ID (1-N). This subtest has two parts.

First, each CPU locks and writes data to its data chunk. Identical data is written for each CPU. Then each CPU reads and compares the information on its data chunk with that of another CPU.

If two CPUs do not confirm consistent data, the test fails and returns an error message. If that happens, testing stops and this test is run again in verbose mode to return more detailed information.

Cache Consistency Requires two or more processors to access and write to the same physical address. This test verifies that a change in physical address by one processor is confirmed by another.

If two processors do not confirm consistent data, the test continues to run, but the Pass Count in the SunVTS status window stops incrementing. If this happens, stop testing and run the test again in verbose mode for a more detailed picture of the problem.

| **Table 37.1** mptest Options (Continued) |
|-------------------|-----------------------------------------------|
| **mptest Options** | **Description** |
| Data I/O | Requires two or more threads, each of which locks onto one of the processors. Each processor, in turn, writes data to a temporary file that has been mapped to the physical address. The modified data is immediately read by other processors being tested. This test hangs and fails if the processors do not recognize the expected data. |
| Shared Memory | A shared memory buffer is divided into a number of contiguous chunks, one for each of the CPUs participating in the test. Each CPU is assigned a unique chunk based upon its ID (1-N). This subtest has two parts. First, each CPU locks and writes data to its data chunk. Identical data is written for each CPU. Then each CPU reads and compares the information on its data chunk with that of another CPU. If two CPUs do not confirm consistent data, the test fails and returns an error message. If that happens, testing stops and this test is run again in verbose mode to return more detailed information. |
| Cache Consistency | Requires two or more processors to access and write to the same physical address. This test verifies that a change in physical address by one processor is confirmed by another. If two processors do not confirm consistent data, the test continues to run, but the Pass Count in the SunVTS status window stops incrementing. If this happens, stop testing and run the test again in verbose mode for a more detailed picture of the problem. |
**mpptest Test Modes**

**TABLE 37-2 mpptest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Checks the current processors on the system with the original processor mask. An error is reported if the two values do not match. The original processor mask is set during probing, which shows the processors on system during the probe. The status of each selected processor is checked by processor_bind.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes ( Offline)</td>
<td>This test mode verifies that the current processor mask is the same as that from the command line, or the same as that from the GUI/TTYUI.</td>
</tr>
</tbody>
</table>

**mpptest Command-Line Syntax**

```
/opt/SUNWvts/bin/mpetest standard_arguments
-o M=0+1+2+3...,NL,ND,NS,NC,omask=hexidecimal_number
```

**TABLE 37-3 mpptest Command-Line Syntax**

<table>
<thead>
<tr>
<th>Arguments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M=0+1+2+3...</td>
<td>Use 0, 1, 2 to specify the processors to test.</td>
</tr>
<tr>
<td>NL</td>
<td>Disables the Lock/Unlock subtest.</td>
</tr>
<tr>
<td>ND</td>
<td>Disables the Data I/O subtest.</td>
</tr>
<tr>
<td>NS</td>
<td>Disables the Shared Memory subtest.</td>
</tr>
<tr>
<td>NC</td>
<td>Disables the Cache Consistency subtest.</td>
</tr>
<tr>
<td>omask=hexidecimal_number</td>
<td>Original mask of processors. Bit 0 represents processor 0 and bit 1 represents processor 1. For example, 03333320.</td>
</tr>
</tbody>
</table>
**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 “32-Bit and 64-Bit Tests” on page 13.
Network Hardware Test (nettest)

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

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

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

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

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

- Random test—sends out 256 packets with random data length and random data.
- Incremental test—sends out packets with length from minimum to maximum packet size using incremental data. (Minimum and maximum values differ for each device.)
- Pattern test—sends 256 packets of maximum length, where each packet contains one test pattern, and all byte patterns (0 to 0xFF hex) are used. That is, the first packet contains pattern 0, the second packet contains pattern 1, and so on, until the last packet pattern of 0xFF.
**Note** — nettest is a scalable test. However, the maximum number of networked devices allowed on a system is 255, and the number of instances for each device is limited to 200. So, if you start the SunVTS exerciser using the -i option to specify a default number of instances for all tests, nettest cannot assign more than 200 instances per each networked device.

**nettest Options**

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
**FIGURE 38-1** nettest Test Parameter Options Dialog Box
The Configuration section specifies the host name, host ID, host address, and domain name of the system being tested.

**TABLE 38-1  nettest Options**

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

**nettest Test Modes**

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

**TABLE 38-2  nettest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Checks whether the device is connected. It searches through all the network interfaces for a specified device name. If nettest does not find the device connected, the test fails; otherwise it returns: device is connected.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Performs all three tests (Random test, Incremental test, and Pattern test) sequentially. It allows you to specify options that will perform heavy stress testing.</td>
</tr>
</tbody>
</table>
nettest Command-Line Syntax

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

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

**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 “32-Bit and 64-Bit Tests” on page 13.
CHAPTER 39

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), and ce (GigaSwift Ethernet). 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 will not be the same.

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

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 if the network interface is connected to a live network, and
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 “Twisted-Pair Ethernet (TPE) Loopback Cable For Fast Ethernet” on page A379 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 “TPE Loopback Cable For Gigabit Ethernet” on page A379 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.
Refer to TABLE 39-1 for test parameter descriptions.

**TABLE 39-1 netlbtest Options**

<table>
<thead>
<tr>
<th>netlbtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>Specifies the Port Address, Host ID, and Domain Name of the system under test.</td>
</tr>
<tr>
<td>Total Packets</td>
<td>Specifies the total number of the packets to send. The default number of packets is 1000.</td>
</tr>
<tr>
<td>Packet size</td>
<td>Determines the size (in bytes) of the packets to be transmitted. 60 &lt;= packet size &lt;= 1514. The default packet size is 1000 bytes.</td>
</tr>
</tbody>
</table>
Since `netlbtest` requires a loopback connector, it can only be selected when Intervention mode is enabled.

### `netlbtest` Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of subtests. It is assumed that the host is not connected to the network through the intended test device(s).</td>
</tr>
</tbody>
</table>

### `netlbtest` Command-Line Syntax

```
/opt/SUNWvts/bin/netlbtest standard_arguments
-o dev=device, tpkts=n, pksz=pkt_size, lb=Internal, warn=Disable
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the device to test such as ge0 or eri0.</td>
</tr>
<tr>
<td>tpkts=n</td>
<td>[1...100000], count of packets to loopback.</td>
</tr>
</tbody>
</table>
TABLE 39-3  netlbtest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pksz=</td>
<td>pkt_size</td>
</tr>
<tr>
<td>lb=</td>
<td>Internal</td>
</tr>
<tr>
<td>warn=</td>
<td>Disable</td>
</tr>
</tbody>
</table>

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 “32-Bit 
and 64-Bit Tests” on page 13.
PCMCIA is a technology that provides small, easy to use peripheral devices. PCMCIA stands for Personal Computer Memory Card International Association. It is a PC Card standard for mobile computing I/O cards. These cards range from memory, FAX/modems, serial I/O, SCSI, video, sound, and so on.

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

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

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

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

**pcsertest Options**

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

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional ( Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests.</td>
</tr>
</tbody>
</table>

**pcsertest** Command-Line Syntax

```
/opt/SUNWvts/bin/pcsertest standard_arguments -o dev=device_name, type=card_type, baudrate=speed, numchars=n
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the device name (for example, dev=pc0 and dev=pc1)</td>
</tr>
<tr>
<td>type=card_type</td>
<td>Specify one of the two card types for the device (type=serial or type=modem). You do not need to specify the type if the device is a modem, since modem is the default card type.</td>
</tr>
<tr>
<td>baudrate=speed</td>
<td>Specifies the communication speed. Specify one of the following: 9600, 19200, 38400, 57600. The default is 9600.</td>
</tr>
<tr>
<td>numchars=n</td>
<td>Specifies the number of characters to use for external loopback testing of the serial socket card. By default, this is set to 256 characters. This option is applicable only to serial socket cards and ignored for the modem card.</td>
</tr>
</tbody>
</table>
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 “32-Bit
and 64-Bit Tests” on page 13.
SPARCstorage Array Controller Test (plntest)

plntest checks the functionality of the controller board on the SPARCstorage™ Array. The SPARCstorage Array (SSA) is a large disk storage I/O subsystem capable of housing up to 30 SCSI hard drives. The SSA communicates with a host system over a fiber-optic link provided by an SBus-based host adapter card in the host system and the corresponding SSA controller board hardware.

The SSA controller card is an intelligent, CPU-based board with its own memory and ROM-resident software. In addition to providing a communications link to the disk drives, it also buffers data between the host system and disk drives in its nonvolatile RAM (NVRAM). For data to go from the host to a particular disk, it must first be successfully transferred to this NVRAM space.

The host machine, SBus host adapter card, Fibre Channel connection, and the SSA controller board must be working properly to perform this data transfer operation. By verifying and stressing this operation, plntest can isolate failures on the SSA disk drives from failures on the SSA controller board.

**Note** – disktest transfers data on the SSA disk drives over the same path mentioned above. However, disktest does not transfer data as quickly as plntest.

plntest Controller Test

The plntest exercises the hardware and software by invoking SCSI read buffer commands of various sizes to the NVRAM. These operations exercise the host Fibre Channel hardware, the SSA Fibre Channel hardware, the SSA resident management
software, and the hardware component interaction on the SSA controller card (all components except the SCSI devices). In addition, the \texttt{plntest} reports failure of the fan module and the NVRAM battery module of the SPARC Storage Array.

## Probing for SSA Controller Devices

Unlike most other hardware devices, the SSA controller card does not have a logical device name (one you would find in the \texttt{/dev} directory). Therefore, the SSA controller card is identified by its longer, physical device name.

\textbf{Note} – The physical device name of the SSA controller card cannot be used to run \texttt{plntest}.

When running \texttt{plntest} from the command line, the physical device name of the SSA controller card cannot be used, so a logical name must be specified. ANSI standards require commas as delimiters between items. Since the physical name of the SSA controller contains embedded commas, if you use a physical name that contains commas as a command-line option, \texttt{plntest} misinterprets the option.

There are two ways that you can create a logical name:

- Run the SunVTS kernel (\texttt{vtsk}), which automatically creates a logical name entry for the SSA controller under the \texttt{/dev} directory, such as:

  ```
  # /dev/ssaXX , where XX represents the decimal number of the controller
  ```

  Use this name as the parameter for the \texttt{dev=} option of \texttt{plntest}.  

  ```
  ```
Manually make a soft link from the actual physical device name to a logical name of your choice (under /dev). Use this name as the parameter for the `dev=` option of `plntest`, as shown in the following example:

```
machine# ./plntest -o "?"
1: /devices/io-unit@f,e3200000/sbi@0,0/SUNW,soc@1,0/SUNW,
   pln@0c0d,0e0f0102:ctlr
2: /devices/io-unit@f,e0200000/sbi@0,0/SUNW,soc@3,0/SUNW,
   pln@0c0d,0e0f0102:ctlr

machine# ln -s /devices/io-unit@f,e3200000/sbi@0,0/SUNW,soc@1,0/SUNW,
   pln@0c0d,0e0f0102:ctlr /dev/ssa1
machine# /opt/SUNWvts/bin/plntest dev=/dev/ssa1
```

### plntest 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.
Configuration lists the names of all the logical disk drives (both single and grouped) that are attached to the SPARCstorage Array controller board. `plntest` is not a scalable test.

**Note** – If no disks are present, none is displayed under the Attached Disks heading.
plntest Test Modes

### TABLE 41-1 plntest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>In this mode, plntest checks the state of the battery module and the fan module of the SPARCstorage Array.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>plntest checks the state of the battery module and the fan module of the SPARCstorage Array. The plntest issues a SCSI command, READBUFFER, causing the entire NVRAM to be read. This test uses different transfer buffer sizes.</td>
</tr>
</tbody>
</table>

plntest Command-Line Syntax

```
/opt/SUNWvts/bin/plntest standard_arguments -o ?,dev=device_name,x
```

### TABLE 41-2 plntest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Probes the system for valid SSA controller devices and prints them to screen (see “Probing for SSA Controller Devices” on page 41246).</td>
</tr>
<tr>
<td>dev=device_name</td>
<td>Specifies the physical path name of the SSA controller card to be tested. This argument must be included when running plntest from the command line, unless the ? argument is used.</td>
</tr>
<tr>
<td>x</td>
<td>Probes the specified SSA controller card for the single and grouped disks attached to the controller card, and prints their logical names to the screen.</td>
</tr>
</tbody>
</table>

Note: The dev=device_name option must be specified for this option to work.

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 “32-Bit and 64-Bit Tests” on page 13.
Physical Memory Test (pmemtest)

The pmemtest checks the physical memory of the system. The pmemtest locates parity errors, hard and soft error correction code (ECC) errors, memory read errors, and addressing problems. The pseudo driver mem(7) is used to read the physical memory.

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

pmemtest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 42-1  pmemtest Test Parameter Options Dialog Box

TABLE 42-1  pmemtest Options

<table>
<thead>
<tr>
<th>pmemtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>Shows the total amount of physical memory, rounded up to the nearest megabyte, probed by the SunVTS kernel.</td>
</tr>
<tr>
<td>Amount of Memory</td>
<td>Specifies the percent of the physical memory to be tested. The default is 33%.</td>
</tr>
</tbody>
</table>
### pmemtest Options

<table>
<thead>
<tr>
<th>pmemtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECC Report Threshold</td>
<td>Determines how many correctable ECC errors occurred in the elapsed time before pmemtest reports a test failure. A value of zero results in no report of any correctable ECC errors. The default is 2. This option is only available on UltraSPARC™ systems.</td>
</tr>
<tr>
<td>Section ID</td>
<td>When set to -1, pmemtest will test one memory section in each pass, automatically testing each subsequent memory section as testing progresses. When set to a number other than -1, only the section specified will be tested. A section is defined by the pass and instance number settings. This option is only available on UltraSPARC systems.</td>
</tr>
<tr>
<td>Instance</td>
<td>The number of copies of pmemtest to run simultaneously on the memory you are testing.</td>
</tr>
</tbody>
</table>
pmemtest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Test</td>
<td>Yes</td>
<td>In this mode, one percent of the memory is read. pmemtest also informs the user how much physical memory is available. For sun4m, sun4u, and UltraSPARC servers, the test reports the ECC errors that have occurred since it was last invoked. The test reports ECC errors for a particular CPU or memory board when physical mapping is selected, otherwise it provides the SIMM number of the ECC memory error.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>In Functional test mode, the amount of memory to be read can vary. For UltraSPARC servers, this test mode reports all ECC errors that have occurred since the system was brought online</td>
</tr>
</tbody>
</table>

pmemtest Command-Line Syntax

```
/opt/SUNWvts/bin/pmemtest standard_arguments
-o size=[1-100],dev=device_name,threshold=report_threshold
,bindinfo=number,section=section_id
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>size=1-100</td>
<td>Specifies the percentage of memory to be tested. The default is 33%.</td>
</tr>
<tr>
<td>dev=device_name</td>
<td>Specifies the device to test, for example, mem.</td>
</tr>
</tbody>
</table>
### Table 42-3: `pmemtest` Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>threshold=</code></td>
<td>Determines how many correctable ECC errors occur before they are reported as an error causing <code>pmemtest</code> to report a failure. A value of zero results in no report of any correctable ECC errors. The default is 2. This option is only available on UltraSPARC systems.</td>
</tr>
<tr>
<td><code>bdinfo=</code></td>
<td>For UltraSPARC servers, indicates board number information. For example, if board 0 and board 5 have memory and you want the test to read the memory on both boards, then <code>bdinfo=33 (2**5+2**0)</code>.</td>
</tr>
<tr>
<td><code>section=</code></td>
<td>When set to -1, <code>pmemtest</code> will test one memory section in each pass, automatically testing each subsequent memory section as testing progresses. When set to a number other than -1, only the section specified will be tested. A section is defined by the pass and instance number settings. This option is only available on UltraSPARC systems.</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Qlogic 2202 Board Test (qlctest)

qlctest is made up of several subtests that test the functions of the Qlogic 2202 FC/AL Crystal. Unlike the earlier single-port Q2100 board, the Q2202 is a two-port board which has greater diagnostic support.

This test is not scalable.

**Note** – Do not run customer data while running qlctest, as the test will take priority over customer data requests. The customer will be unable to access data while qlctest is running.

**Note** – Do not run other tests while qlctest is running. qlctest may cause other tests to fail.

**Note** – qlctest is an intervention mode test. No subtests can be selected unless intervention is set.

### qlctest Subtests

There are nine possible subtests to run in intervention and functional modes:

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

The external loopback test is an intervention test. To test the fibre loop, leave the QLC port attached to the storage. In the Test Parameters Options dialog box, set the "Test if Connected to Storage" option to "Yes". To test the Qlogic 2202 board alone, connect a loopback cable to the QLC port. This cable can be made by taking a regular cable and splitting it apart. Then loop the transmitter side of the port to the receiver side of the port.

For subtest descriptions, see TABLE 43-1 on page 259.

qlctest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 43-1 qlcitest Test Parameter Options Dialog Box

TABLE 43-1 qlcitest Options

<table>
<thead>
<tr>
<th>qlcitest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fcode revision check</td>
<td>Retrieves the fcode revision string. A core subtest that is always run but not shown in the Options dialog box.</td>
</tr>
<tr>
<td>Firmware revision check</td>
<td>Retrieves the firmware revision string. A core subtest that is always run but not shown in the Options dialog box.</td>
</tr>
<tr>
<td>Board revision check</td>
<td>Retrieves the board revision levels. A core subtest that is always run but not shown in the Options dialog box.</td>
</tr>
<tr>
<td>Test if Connected to Storage</td>
<td>Runs qlcitest while connected to storage. Default value is No.</td>
</tr>
</tbody>
</table>
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.

Mailbox Loopback subtest Loads a series of registers into the input mailboxes on the card and then reads the output mailboxes and compares results. This verifies that the system side of the card is operating correctly, and that the internal data paths are correct. Run by default, but can be deselected.

Firmware Checksum subtest Runs an internal checksum test on the installed firmware. This verifies that the RISC RAM on the card is fully functional and that the installed firmware is still intact. This test also serves as a quick RAM check of the RISC RAM. Run by default, but can be deselected.

Internal 10-bit Loopback subtest Performs internal loopback test within the host adapter ISP hardware at the 10-bit interface. This test is done with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected via the test parameters menu. Run by default, but can be deselected.

Internal 1-bit Loopback subtest Performs internal loopback test within the host adapter ISP hardware at the 1-bit interface. This test is done with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected via the test parameters menu. Run by default, but can be deselected.

External Loopback subtest Performs an external loopback test. This test is done with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected via the test parameters menu. This is an intervention test, because a loopback cable is needed from the transceiver to the receiver of the QLC port when testing this port by itself. This subtest can also test the entire fibre channel loop when the loop is connected to the storage to be tested. Not run by default, but can be deselected.

<table>
<thead>
<tr>
<th>qlctest Options</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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. |
| Mailbox Loopback subtest | Loads a series of registers into the input mailboxes on the card and then reads the output mailboxes and compares results. This verifies that the system side of the card is operating correctly, and that the internal data paths are correct. Run by default, but can be deselected. |
| Firmware Checksum subtest | Runs an internal checksum test on the installed firmware. This verifies that the RISC RAM on the card is fully functional and that the installed firmware is still intact. This test also serves as a quick RAM check of the RISC RAM. Run by default, but can be deselected. |
| Internal 10-bit Loopback subtest | Performs internal loopback test within the host adapter ISP hardware at the 10-bit interface. This test is done with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected via the test parameters menu. Run by default, but can be deselected. |
| Internal 1-bit Loopback subtest | Performs internal loopback test within the host adapter ISP hardware at the 1-bit interface. This test is done with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected via the test parameters menu. Run by default, but can be deselected. |
| External Loopback subtest | Performs an external loopback test. This test is done with data sourcing from the system memory and going to the system memory. The desired data pattern, transfer length, and iteration count can be selected via the test parameters menu. This is an intervention test, because a loopback cable is needed from the transceiver to the receiver of the QLC port when testing this port by itself. This subtest can also test the entire fibre channel loop when the loop is connected to the storage to be tested. Not run by default, but can be deselected. |
### TABLE 43-1 qlctest Options

<table>
<thead>
<tr>
<th>qlctest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loopback Transfer Count</td>
<td>Controls the packet size used in the internal 10-bit, internal 1-bit, and external loopback tests. Default value is 0x10000.</td>
</tr>
<tr>
<td>Loopback Iteration Count</td>
<td>Sets the number of times to loop the internal 10-bit, internal 1-bit, and external loopback tests. Default value is 10.</td>
</tr>
<tr>
<td>Loopback Data Pattern</td>
<td>Selects the data pattern to loop for the internal 10-bit, internal 1-bit, and external loopback tests. Default value is 0x7e7e7e7e.</td>
</tr>
</tbody>
</table>
qlctest Test Modes

TABLE 43-2 qlctest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Opens and closes the QLC port.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full set of tests.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

qlctest Command-Line Syntax

```
/opt/SUNWvts/bin/qlctest standard_arguments
-v -o dev=device name, run_connect=Yes | No, selftest=Enable | Disable, mbox=Enable | Disable, checksum=Enable | Disable, ilb_10=Enable | Disable, ilb=Enable | Disable, elb=Enable | Disable, xcnt=0xtransfer_count, icnt=iteration_count, lbfpattern=0xpatter
```

TABLE 43-3 qlctest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev</td>
<td>The name of the device to test.</td>
</tr>
<tr>
<td>run_connect=Yes</td>
<td>No</td>
</tr>
<tr>
<td>selftest=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>mbox=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>checksum=Enable</td>
<td>Disable</td>
</tr>
</tbody>
</table>
TABLE 43-3  qlctest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ilb_10=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>ilb=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>elb=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>xcnt=0xtransfer_count</td>
<td>Controls the packet size to be transferred, for example, 0x1000. Default value is 0x10000.</td>
</tr>
<tr>
<td>icnt=iteration_count</td>
<td>Controls the number of times the loopback test will run, for example, 100. Default value is 10.</td>
</tr>
<tr>
<td>lbfpattern=0xpattern</td>
<td>Lists the data pattern to loop, for example, 0x7E7E7E7E. Default value is 0x7E7E7E7E.</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Remote System Control (rsctest)

The rsctest exercises the Remote System Control (RSC) feature, which is integrated on the Sun Enterprise 250 as well as the next-generation RSC 2.0 plug-in card introduced with the Sun Fire 280R line.

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

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

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

rsctest is not scalable.

rsctest Subtests

The rsctest will present different subtests and options based on which revision of the RSC hardware it is testing.
The subtests common to both RSC 1.0 and 2.0 include:

TABLE 44-1  Subtests for both RSC 1.0 and 2.0

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

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

TABLE 44-2  Subtests for RSC 2.0 Only

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

The subtests call test modlets that are written in the native Real Time Operating System (RTOS) that resides in the RSC firmware. The rsctest subtests execute the test modlets, passes parameters, and retrieves results from the RSC using a test protocol on the host to RSC internal serial lines.
rsctest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 44-1  rsctest Test Parameter Options Dialog Box
<table>
<thead>
<tr>
<th>rsctest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enet test</td>
<td>Enables or disables RSC Ethernet testing.</td>
</tr>
<tr>
<td>Data Pattern Type</td>
<td>Selects Sequential, Random, or both types of data patterns.</td>
</tr>
<tr>
<td>Packet Size</td>
<td>Defines the size of each data packet to be sent for all tests.</td>
</tr>
<tr>
<td>Num Packets</td>
<td>Specifies the number of data packets to send in one test loop.</td>
</tr>
<tr>
<td>Target Host</td>
<td>Specifies the IP address of a host to use for the ping test.</td>
</tr>
<tr>
<td>Enet Test Type</td>
<td>Selects any or all Internal, External, or ping tests.</td>
</tr>
<tr>
<td>Flash test</td>
<td>Enables or disables the flash checksum test.</td>
</tr>
<tr>
<td>SEEPROM test</td>
<td>Enables or disables the SEEPROM checksum test.</td>
</tr>
<tr>
<td>FRU SEEPROM test</td>
<td>Enables or disables the FRU SEEPROM checksum test (RSC 2.0 only).</td>
</tr>
<tr>
<td>TOD test</td>
<td>Enables or disables the Time Of Day test.</td>
</tr>
<tr>
<td>I2C test</td>
<td>Enables or disables the I2C test (RSC 2.0 only).</td>
</tr>
<tr>
<td>Serial test</td>
<td>Enables or disables the RSC serial test.</td>
</tr>
<tr>
<td>Data Size</td>
<td>Defines the data size to be sent.</td>
</tr>
<tr>
<td>Loopback Type</td>
<td>Selects Internal, External, or both. External requires a loopback plug.</td>
</tr>
<tr>
<td>Data Pattern Type</td>
<td>Selects Sequential, Random, or both types of data patterns.</td>
</tr>
<tr>
<td>Serial Test Type</td>
<td>Selects serial ports to be tested, u to u, c to c, or d to d.</td>
</tr>
<tr>
<td>TTYU_Baud</td>
<td>Select a fixed baud rate or all baud rates for testing the ttyu port. The valid baud rates under TTYU_Baud are: ALL, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 76800, 115200. The default is 9600.</td>
</tr>
<tr>
<td>Modem Test</td>
<td>Used to Enable or Disable the RSC PCMCIA modem test (RSC 2.0 only).</td>
</tr>
</tbody>
</table>
rsctest Test Modes

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

**TABLE 44-4  rsctest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Reports the status of the RSC.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Tests the RSC’s Ethernet, flash, SEEPROM, and serial devices.</td>
</tr>
<tr>
<td>Offline</td>
<td></td>
<td>All tests use internal modes as defaults. The rsctest will not run the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>serial test on ttyc if the console has been redirected to the RSC.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The ttyu tests will not run if there is an open login on ttyu.</td>
</tr>
</tbody>
</table>

rsctest Command-Line Syntax

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


**TABLE 44-5  rsctest Command-Line Syntax**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enet=enable</td>
<td>disable</td>
</tr>
<tr>
<td>epatttype=seq+rand</td>
<td>Predefined pattern options used for Enet test.</td>
</tr>
<tr>
<td>esize=packet_size</td>
<td>Data size for each packet in the Enet test.</td>
</tr>
<tr>
<td>epkts=number_packets</td>
<td>Number of packets to send for Enet test.</td>
</tr>
<tr>
<td>target=IP_address</td>
<td>IP address of target system for Enet ping test.</td>
</tr>
<tr>
<td>etest=Internal+External+ Ping</td>
<td>Selects any or all Internal, External, or ping tests.</td>
</tr>
</tbody>
</table>
TABLE 44-5  
rsctest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>flash=enable</td>
<td>disable</td>
</tr>
<tr>
<td>seeprom=enable</td>
<td>disable</td>
</tr>
<tr>
<td>fruseeprom=E/D (RSC 2.0 ONLY)</td>
<td>Enables or disables RSC FRU SEEPROM checksum test.</td>
</tr>
<tr>
<td>tod=E/D (RSC 2.0 ONLY)</td>
<td>Enables or disables RSC Time Of Day test.</td>
</tr>
<tr>
<td>i2c=E/D (RSC 2.0 ONLY)</td>
<td>Enables or disables RSC i2c test.</td>
</tr>
<tr>
<td>serial=enable</td>
<td>disable</td>
</tr>
<tr>
<td>sdatsize=data_size</td>
<td>Data size for the rsc serial tests.</td>
</tr>
<tr>
<td>slb=Internal</td>
<td>External</td>
</tr>
<tr>
<td>spatttype=seq+rand</td>
<td>Predefined pattern options used for RSC serial test.</td>
</tr>
<tr>
<td>stest=u_u+c_c+d_d</td>
<td>Defines port and configuration to use for RSC serial test.</td>
</tr>
<tr>
<td>ttyu_baud=ALL</td>
<td>specific_baud</td>
</tr>
<tr>
<td>rscmodem=E/D</td>
<td>Enables or disables the RSC PCMCIA modem test.</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Serial Asynchronous Interface (PCI) Test (saiptest)

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

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

saiptest Hardware Requirements

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

Note – You must run the saiptest in Intervention mode.

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

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

- PCI-based SPARC system with a PCI slot
- Serial asynchronous interface card, installed in one of the PCI slots
The following hardware is also required to run other SunVTS Serial Asynchronous Interface tests:

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

**saiptest Options**

To reach the 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.
The Configuration section of the Options dialog box displays the asynchronous serial ports available for the SAI board. Table 4-1 shows the available ports.

<table>
<thead>
<tr>
<th>Board Number</th>
<th>Board Device</th>
<th>Serial Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>saip0</td>
<td>term/a000-a007</td>
</tr>
<tr>
<td>1</td>
<td>saip1</td>
<td>term/b000-b007</td>
</tr>
<tr>
<td>2</td>
<td>saip2</td>
<td>term/c000-c007</td>
</tr>
<tr>
<td>3</td>
<td>saip3</td>
<td>term/d000-d007</td>
</tr>
</tbody>
</table>
TABLE 45-1 saiptest Asynchronous Serial Ports  (Continued)

<table>
<thead>
<tr>
<th>Board Number</th>
<th>Board Device</th>
<th>Serial Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>saip4</td>
<td>term/e000-e007</td>
</tr>
<tr>
<td>5</td>
<td>saip5</td>
<td>term/f000-f007</td>
</tr>
<tr>
<td>6</td>
<td>saip6</td>
<td>term/g000-g007</td>
</tr>
<tr>
<td>7</td>
<td>saip7</td>
<td>term/h000-h007</td>
</tr>
<tr>
<td>8</td>
<td>saip8</td>
<td>term/i000-i007</td>
</tr>
<tr>
<td>9</td>
<td>saip9</td>
<td>term/j000-j007</td>
</tr>
<tr>
<td>10</td>
<td>saip10</td>
<td>term/k000-k007</td>
</tr>
<tr>
<td>11</td>
<td>saip11</td>
<td>term/l000-l007</td>
</tr>
<tr>
<td>12</td>
<td>saip12</td>
<td>term/m000-m007</td>
</tr>
<tr>
<td>13</td>
<td>saip13</td>
<td>term/n000-n007</td>
</tr>
<tr>
<td>14</td>
<td>saip14</td>
<td>term/o000-o007</td>
</tr>
<tr>
<td>15</td>
<td>saip15</td>
<td>term/p00-p007</td>
</tr>
</tbody>
</table>

TABLE 45-2 saiptest Options

<table>
<thead>
<tr>
<th>saiptest Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Test</td>
<td>Performs internal loopback testing on the SAI card(s) installed in PCI slots. You do not need to attach anything to the card(s) to perform this test.</td>
</tr>
<tr>
<td>25-pin Loopback</td>
<td>Provides full-duplex transmission and full-modem loopback testing of the serial port selected in the Serial Port section of the option menu. Attach the 25-pin loopback plug to the serial port on the serial asynchronous interface patch panel that is being tested. This test cannot be run concurrently with the Echo-TTY option enabled.</td>
</tr>
<tr>
<td>Echo-TTY</td>
<td>Checks the proper operation of the serial port selected in the Serial Port selection of the option menu by echoing characters typed on a TTY terminal keyboard to the TTY terminal screen. The characters you type on your TTY keyboard display on the TTY screen. Note: A TTY connection to the serial asynchronous interface serial port requires corresponding character size setup. For example, if a TTY attachment is running with 8-bit character size, then the Char Size saiptest option should be set to 8-bits. If you do not type any characters within two minutes, this test times-out.</td>
</tr>
</tbody>
</table>
Chapter 45 Serial Asynchronous Interface (PCI) Test (saiptest)

saiptest Test Modes

**TABLE 45-3 saiptest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests.</td>
</tr>
</tbody>
</table>

saiptest Command-Line Syntax

```
/opt/SUNWvts/bin/saiptest standard_arguments -o dev=device_name, M=test_mode, B=baud_rate, Size=character_size, Stop=#of_stop_bits, Parity=parity, F=flow_control, Data=test_pattern, sp=serial_port, tout=time_out
```
### TABLE 45-4 saiptest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| dev=device_name | Specifies the asynchronous serial ports of the PCI card slots tested. Since there is no default, you must type a device name—either a board(sai0-16) or an individual port (term/x00-term/x007, where x is a-p):  
  • sai0 = the 8 asynchronous serial ports in the first card  
  • sai1 = the 8 asynchronous serial ports in the second card  
  • sai2 = the 8 asynchronous serial ports in the third card  
  • sai3 = the 8 asynchronous serial ports in the fourth card  
  • sai4 = the 8 asynchronous serial ports in the fifth card  
  • sai5 = the 8 asynchronous serial ports in the sixth card  
  • sai6 = the 8 asynchronous serial ports in the seventh card  
  • sai7 = the 8 asynchronous serial ports in the eighth card  
  • sai8 = the 8 asynchronous serial ports in the ninth card  
  • sai9 = the 8 asynchronous serial ports in the tenth card  
  • sai10 = the 8 asynchronous serial ports in the eleventh card  
  • sai11 = the 8 asynchronous serial ports in the twelfth card  
  • sai12 = the 8 asynchronous serial ports in the thirteenth card  
  • sai13 = the 8 asynchronous serial ports in the fourteenth card  
  • sai14 = the 8 asynchronous serial ports in the fifteenth card  
  • sai15 = the 8 asynchronous serial ports in the sixteenth card  
  or  
  /dev/term/x00m  
  Where x is a-p and m is 0 to 7 (any of the asynchronous serial ports in PCI card slots). |
| M=test_mode    | Specifies Internal, 25_pin_loopback, or Echo_TTY test mode.                                                                                   |
| B=baud_rate    | Sets the baud rate to 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400; the default is 9600.                                             |
| Stop=#of_stop_bits | Toggles the number of stop bits between 1 or 2. The default is 1.                                                                         |
| Size=character_size | Sets the character size as an integer between 5 and 8.                                                                                |
| Parity=parity  | Specifies the parity as none, odd, or even. The default is none.                                                                          |
| F=flow_control | Specifies flow control as xonoff, rtscts, or both.                                                                                        |
| Data=test_pattern | Specifies test pattern as 0x55555555, 0xAAAAAAAAA, or random.                                                                          |
| sp=serial_port | Specifies the terminal and asynchronous serial port number, such as term/a00n (sp=n).                                                   |
| tout=time_out  | Specifies the number of seconds until the test times out. The default is 120 seconds.                                                     |
**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 “32-Bit and 64-Bit Tests” on page 13.

**Note** – The `saipptest` error messages are generated when the SunVTS Serial Asynchronous Interface discovers errors. The error descriptions that appear in the VTS interface identify probable causes for the card or test failure, and identify the Field Replaceable Unit (FRU) and recommended action, if possible. The two FRUs under test are: the Serial Asynchronous Interface card, and the patch panel with the cable attached.
Sun Enterprise Cluster 2.0 Network Hardware Test (scitest)

scitest verifies the functionality of the Sun Enterprise Cluster 2.0 by checking the networking hardware. For this test to be meaningful, the cluster must already be configured before the test is run. For details on how to configure the cluster, refer to Sun Enterprise Cluster 2.0 Hardware Site Preparation, Planning, and Installation Guide.

scitest reads the /etc/sma.ip file to determine the target nodes in the cluster. scitest mainly uses the Internet Control Message Protocol (ICMP) to test the connections between cluster nodes.

After finding the cluster nodes (targets), scitest performs the following tests:
- Random test—sends out 256 packets with random data length and random data.
- Incremental test—sends out packets with length from minimum to maximum packet size using incremental data.
- Pattern test—sends 256 packets of maximum length, where each packet contains one test pattern and all byte patterns (0 to 0xFF) are used.

Note – scitest is a scalable test. The maximum number of instances is two per SCI card.

scitest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 46-1 scitest Test Parameter Options Dialog Box

TABLE 46-1 scitest Options

<table>
<thead>
<tr>
<th>scitest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Host</td>
<td>Not used.</td>
</tr>
<tr>
<td>Receive Timeout field</td>
<td>Specify a number between 0–600 seconds. The default is 120 seconds.</td>
</tr>
<tr>
<td>Number of Retries field</td>
<td>The number of retries before an error is flagged. Specify a number between 0–128.</td>
</tr>
<tr>
<td>Print Warning</td>
<td>Disabled by default. Choose Enable to see warning errors, such as retry on timeout errors.</td>
</tr>
</tbody>
</table>
scitest Test Modes

Connection, Functional, and Online modes are supported by scitest. Different test schemes are performed on the network device based on the mode selected.

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>scitest checks if the device is connected. It searches through all the network interfaces for the specified device name. If scitest finds the device not connected the test fails, otherwise, it returns device is connected.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>scitest performs all three tests (Random test, Incremental test, and Pattern test) sequentially. It allows you to specify an option in such a way that scitest performs a very stressful test.</td>
</tr>
</tbody>
</table>

scitest Command-Line Syntax

```
/opt/SUNWvts/bin/scitest standard_arguments -o dev=interface, test=type, packets=n, pattern=hex, delay=seconds, timeout=seconds, retry=n, warn
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=interface</td>
<td>Network interface name. The default value is le0 for Ethernet networks.</td>
</tr>
<tr>
<td>test=type</td>
<td>The test type. Specify random, increment, or pattern. The default value is random+increment+pattern to run.</td>
</tr>
<tr>
<td>packets=n</td>
<td>Number of random/pattern packets. The default is 256.</td>
</tr>
<tr>
<td>pattern=hex</td>
<td>Specifies a data pattern in hexadecimal form. The default is all patterns from 0 to 0xff.</td>
</tr>
<tr>
<td>delay=seconds</td>
<td>Indicates the time between subtests in seconds. The default is 30 seconds.</td>
</tr>
</tbody>
</table>
### TABLE 46-3  scitest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>timeout=seconds</code></td>
<td>Indicates the number of seconds to wait before a timeout. The default is 20 seconds.</td>
</tr>
<tr>
<td><code>retry=n</code></td>
<td>Indicates the number of test timeout retries. The default is three retries.</td>
</tr>
<tr>
<td><code>warn</code></td>
<td>When enabled, prints warning messages.</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Smart Card Test \((sctest)\)

The Smart Card test \((sctest)\) consists of a series of subtests to exercise smart cards and smart card readers. It exercises the Java\textsuperscript{TM} card terminal reader driver, the Solaris I2C SCM device driver, and the serial device driver.

\(sctest\) tests only the internal SC reader. \(sctest\) is only run as an intervention test. Users need to initialize SC card and Internal SC reader to run \(sctest\).

**Note** – Users should initialize the smart card with commands or GUI provided by OS. This initialization information can be found in the Smart Card User Administration Guide at docs.sun.com.

![Caution!]

**Caution** – Do not use a smart card with real data for \(sctest\). Allocate a scratch smart card for testing purposes only. Functions of \(sctest\) will delete any existing data on the card.

**Note** – Do not run \(sctest\) and \(sptest\) at the same time. Test results will conflict if \(sptest\) is enabled when \(sctest\) is run. (Because it is an intervention test, \(sptest\) is disabled by default.)

To select \(sctest\), click the Intervention button in the Select Devices panel.

\(sctest\) Test Requirements

The following five requirements must be met before running \(sctest\):

- You must enable the SunVTS intervention mode.
- You must have a scratch smart card ready.
- You must configure the card reader, if it has not been configured already. The card reader can be configured with the GUI provided by the OS (under Workspace Menu-Tools-Smart card).
- You must initialize the test card with GUI provided by the OS (under Workspace Menu-Tools-Smart card).
- You must run the following command to personalize the card:

```bash
# /usr/bin/smartcard -c init -A A0000000620304000 -P '$$$$java' \  
user=guest application=AAA password=guestwho
```

Once you have fulfilled these five requirements, you are ready to run `sctest`. 
**sctest Subtests**

<table>
<thead>
<tr>
<th>Subtest</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card Presence Detect test</td>
<td>Checks whether a card is present in the reader. Place the card in the reader to begin the test.</td>
</tr>
<tr>
<td>Card Lock/Unlock test</td>
<td>Exercises the functions that lock and unlock a card. Place the card in the reader to begin the test.</td>
</tr>
<tr>
<td>Insertion/Removal test</td>
<td>Tests detection of card insertion and removal. If the card is already in the reader, remove and reinsert when prompted. This test is disabled by default, since it requires user intervention.</td>
</tr>
<tr>
<td>PinCardService test</td>
<td>Waits for a card to be inserted, then performs authentication of the default PIN ($$$$java). Note: this subtest assumes the card has been initialized.</td>
</tr>
<tr>
<td>Simple Authentication test</td>
<td>Waits for a card to be inserted, then performs authentication of the default PIN ($$$$java). Displays the username and password of an example user called AAA. This test only works for cards that have been initialized. This subtest requires the PinCardService test to first validate the PIN on the smart card.</td>
</tr>
<tr>
<td>UserInfoCardService test</td>
<td>Waits for a card to be inserted, then performs authentication of the default PIN ($$$$java). Displays the user information such as the username and password of an example user called AAA. If Write mode is enabled, the test will also create default user information for an example user BBB. If Verbose mode is enabled, this user information will be displayed. Next, the test deletes the user information for BBB. When the test queries for user information for BBB, it expects an exception (since this information is already deleted). This subtest requires that the PinCardService test has validated the PIN on the smart card.</td>
</tr>
</tbody>
</table>

**sctest Options**

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

FIGURE 47-1 sctest Test Parameter Options Dialog Box

TABLE 47-2 sctest Options

<table>
<thead>
<tr>
<th>sctest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>Enables or disables the Write mode for the UserInfoCard Service subtest.</td>
</tr>
<tr>
<td>OCF Based tests</td>
<td>Enables or disables all of the following subtests: Card Presence Detect, Card Lock/Unlock, Insertion/Removal, PinCardService, Simple Authentication, and UserInfoCard.</td>
</tr>
<tr>
<td>Card Removal test</td>
<td>Enables or disables the Insertion/Removal subtest.</td>
</tr>
</tbody>
</table>
sctest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full set of tests.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

sctest Command-Line Syntax

```
/opt/SUNWvts/bin/sctest standard_arguments
-o dev=a|b|i2cscmN, Mode=ReadWrite|ReadOnly, OCFBasedTests=Enable|Disable, CardRemovalTest=Enable|Disable
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=a</td>
<td>b</td>
</tr>
<tr>
<td>Mode=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>OCFBasedTests=Enable</td>
<td>Disable</td>
</tr>
<tr>
<td>CardRemovalTest=Enable</td>
<td>Disable</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
Environmental Sensing Card Test
(sentest)

sentest checks the SCSI Environmental Sensing card (SEN) installed in the SPARCstorage RSM to monitor the enclosure environment. The SEN card monitors the enclosure’s over-temperature condition, fan-failures, power-supply failures, and drive activity.

sentest verifies the SEN card by first setting each control function to a predetermined value, and then reading it back to verify if the value is correct.

sentest verifies the following control functions in the enclosure:

- Alarm (enable/disable)—sentest toggles the alarm to the disable state, then to the enable state.
- Alarm time (0-0xff seconds)—sentest sets the time (from 0 to 4095), then reads it back to verify the time setting.
- Drive fault LED (DL0-DL6)—sentest toggles each LED to its OFF and ON states.

sentest is a nonscalable test. It cannot verify the control function settings correctly if another instance is changing the setting.

sentest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 48-1 sentest Test Parameter Options Dialog Box
sentest Test Modes

sentest supports all three test modes. Each mode performs a different test scheme on the SEN card.

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Checks the device connection by opening the device. If the device does not open, the device is not connected.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Checks three components within the enclosure. It checks alarm enable/disable, alarm time setting, and the drive LEDs. It does not test the power on/off function (only functions whose values can be changed are tested).</td>
</tr>
</tbody>
</table>

**TABLE 48-1**  
<table>
<thead>
<tr>
<th>sentest Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Type</td>
<td>Specifies the test to be performed. Norm test type performs normal testing as alarm enable/disable, alarm time setting, and drive LED on/off testing. Interactive test type reports the current enclosure status.</td>
</tr>
</tbody>
</table>
| Enc_state      | Specifies which subsystem’s status in the enclosure is reported. Default is ALL. This test is only used with the interactive test type and in Functional test. The test options are:  
  • Alarm enable/disable status  
  • Drive present status  
  • Drive LED status  
  • Power modules status  
  • Fan modules status  
  • Over temperature, abs (abnormal, no immediate attention needed), chk (abnormal, immediate attention needed) status  
  • All of the above |
sentest Command-Line Syntax

```
/opt/SUNWvts/bin/sentest standard arguments
-o dev=interface,test=type,enc=component
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=interface</td>
<td>SEN card device name. The default value is ses0.</td>
</tr>
<tr>
<td>test=type</td>
<td>Specifies the test type. Select Norm for normal testing or Inter for interactive testing; the default value is Norm. Possible values are: norm and inter.</td>
</tr>
<tr>
<td>enc=component</td>
<td>Indicates which part of the enclosure status is reported. The default value is ALL. Possible values are: enalm, dp, dl, pm, fan, ovt, and ALL.</td>
</tr>
</tbody>
</table>

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 “32-Bit and 64-Bit Tests” on page 13.
```
Soc+ Host Adapter Card Test
(socaltest)

socaltest aids the validation and fault isolation of the SOC+ host adapter card. In the case of a faulty card, the test tries to isolate the fault to the card, the GBIC module, or the DMA between the host adapter card and the host memory.

**Note** – Do not run socaltest and enatest at the same time, otherwise test failures might occur.

**Note** – Do not run socaltest and disktest at the same time, otherwise test failures might occur.

**Note** – Do not run socaltest with a high system load. Running this test with a large number of instances and concurrency might cause resource limitations that cause this test to fail.

**socaltest Options**

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 49-1  socaltest  Test Parameter Options Dialog Box
TABLE 49-1  socaltest Options

<table>
<thead>
<tr>
<th>socaltest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Loopback test</td>
<td>Checks the host adapter card and the direct memory access (DMA) with the host system. This is accomplished as follows:</td>
</tr>
<tr>
<td></td>
<td>1. A frame is created in the host adapter local memory, sent out through the SOC+ transmitter and internally looped back to the SOC+ receiver. The received data is compared with the original data.</td>
</tr>
<tr>
<td></td>
<td>2. A frame is created in the host adapter local memory, sent out through the SOC+ transmitter and looped back through the SERDES (serialiser-deserialiser) chip on the host adapter card. The received data is compared with the original data.</td>
</tr>
<tr>
<td></td>
<td>3. A frame is created in the host main memory, transferred through the DMA to the host adapter transmitter, looped back within the SOC+ chip, and transferred from the receiver to the host main memory through the DMA. The received frame is compared with the original transmitted frame, which tests the host memory to the host adapter DMA path.</td>
</tr>
<tr>
<td>External Loopback test</td>
<td>The External Loopback test verifies the proper functioning of the GBIC module. A frame is created in the host adapter local memory, sent out and looped back through the external loopback connector attached to the port. If the external loopback test is run together with the internal loopback test, the DMA path is also tested by creating a frame in host main memory, transferring it to the host adapter through the DMA, looping it back through the external loopback connector and transferring the received frame back to the host main memory by DMA.</td>
</tr>
<tr>
<td>Loopback Frame test</td>
<td>Sends out a buffer initialized with the selected pattern and compares it with the looped back frame. It passes if the two compare and fails if they do not.</td>
</tr>
<tr>
<td>Loopback Frame Pattern</td>
<td>List of selectable patterns for the Loopback Frame test.</td>
</tr>
</tbody>
</table>

Note – In addition to the tests described above, socaltest also tests the basic functions of the SOC+ chip, the on-board XRAM, and the host control buffer by invoking the appropriate tests implemented in firmware.
socaltest Test Modes

TABLE 49-2  socaltest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests</td>
</tr>
</tbody>
</table>

Note – You cannot run the Internal and External Loopback tests if the port is connected to a disk array.

socaltest Command-Line Syntax

```
/opt/SUNWvts/bin/socaltest  standard arguments
-o  dev=device name, elb=enabled|disabled, ilb=enabled|disabled, lbf= enable|disable, ptn=pattern
```

TABLE 49-3  socaltest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device name</td>
<td>The name of the socal port to be tested.</td>
</tr>
<tr>
<td>elb=enabled</td>
<td>disabled</td>
</tr>
<tr>
<td>ilb=enabled</td>
<td>disabled</td>
</tr>
<tr>
<td>lbf=enable</td>
<td>disable</td>
</tr>
<tr>
<td>ptn=pattern</td>
<td>Specify the pattern in hexadecimal, for example: ptn= 0x7e7e7e7e</td>
</tr>
</tbody>
</table>

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 “32-Bit and 64-Bit Tests” on page 13.
Serial Parallel Controller Test
*spiftest*

The *spiftest* accesses card components such as the cd-180 and ppc2 chips, and the serial and parallel ports through the serial parallel controller device driver.

**spiftest Hardware Requirements**

Before running the SunVTS system exerciser, make sure you install the cards to be tested and the device driver. You should also reboot your system with the `boot -r` command to reconfigure the system and allow the SunVTS kernel to recognize the new driver.

**Note** – The *spiftest* must be run in Intervention mode.

The following minimum hardware configuration is required to successfully run the Internal test:
- SBus-based SPARC desktop system with an SBus slot
- Serial parallel controller card, installed in one of the SBus slots

The following hardware is also required to run the other SunVTS serial parallel controller tests:
- Serial parallel controller patch panel (part number 540-2007)
- 96-pin loopback plugs (part number 370-1366)
- 25-pin serial loopback plugs (part number 540-1558)
- RS-232 serial cables (part number 530-1685)
- TTY terminal
spiftest 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.

![spiftest Test Parameter Options Dialog Box](image-url)

**FIGURE 50-1** spiftest Test Parameter Options Dialog Box
The Configuration section of the Options dialog box displays the serial ports available for the SPC/S board. The available ports are listed in the table below.

**TABLE 50-1**  
*spiftest Serial Ports for the SPC/S Board*

<table>
<thead>
<tr>
<th>Board Number</th>
<th>Board Device</th>
<th>Serial Ports</th>
<th>Parallel Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>stc0</td>
<td>term/0-7</td>
<td>printers/0</td>
</tr>
<tr>
<td>1</td>
<td>stc1</td>
<td>term/8-15</td>
<td>printers/1</td>
</tr>
<tr>
<td>2</td>
<td>stc2</td>
<td>term/16-23</td>
<td>printers/2</td>
</tr>
<tr>
<td>3</td>
<td>stc3</td>
<td>term/24-31</td>
<td>printers/3</td>
</tr>
<tr>
<td>4</td>
<td>stc4</td>
<td>term/32-39</td>
<td>printers/4</td>
</tr>
<tr>
<td>5</td>
<td>stc5</td>
<td>term/40-47</td>
<td>printers/5</td>
</tr>
<tr>
<td>6</td>
<td>stc6</td>
<td>term/48-55</td>
<td>printers/6</td>
</tr>
<tr>
<td>7</td>
<td>stc7</td>
<td>term/56-63</td>
<td>printers/7</td>
</tr>
</tbody>
</table>

The *spiftest* options are described in the table below.

**TABLE 50-2**  
*spiftest Options*

<table>
<thead>
<tr>
<th>spiftest Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>96-pin Loopback (LB)</td>
<td>Provides data transmission, full-modem loopback, and parallel port loopback testing. You <em>must</em> attach a 96-pin loopback connector (part number 370-1366) to the card before running this test (see Appendix A, “96-Pin Female Loopback Connector” section).</td>
</tr>
<tr>
<td>Internal test</td>
<td>Performs a quick internal check of the serial parallel controller card(s) installed in SBus slots. You do not need to attach anything to the card(s) to perform this test.</td>
</tr>
<tr>
<td>25-pin Loopback (LB)</td>
<td>Provides full-duplex transmission and full-modem loopback testing of the serial port selected in the Serial Port selection of this menu. You <em>must</em> attach the 25-pin Loopback plug to the serial port on the Serial Parallel Controller Patch Panel that is being tested (see Appendix A). This test cannot be run concurrently with the Echo-TTY option enabled.</td>
</tr>
</tbody>
</table>
You can also change the test options by modifying the 
### spiftest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests</td>
</tr>
</tbody>
</table>

### spiftest Command-Line Syntax

```
/opt/SUNWvts/bin/spiftest standard_arguments -o dev=device_name, M=test_mode, Ptr=enable/disable, B=baud_rate, Size=character_size, S=#of_stop_bits, Parity=parity, F=flow_control, Data=test_pattern, sp=serial_port
```

### TABLE 50-3 spiftest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests</td>
</tr>
</tbody>
</table>

### TABLE 50-4 spiftest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| dev=             | Specifies the serial ports in SBus card slots (0-63) being tested. Since there is no default, you must type a board name:  
  - stc0—the 8 serial ports in the first card  
  - stc1—the 8 serial ports in the second card  
  - stc2—the 8 serial ports in the third card  
  - stc3—the 8 serial ports in the fourth card  
  - stc4—the 8 serial ports in the fifth card  
  - stc5—the 8 serial ports in the sixth card  
  - stc6—the 8 serial ports in the seventh card  
  - stc7—the 8 serial ports in the eighth card  |
| M=               | Specifies Internal, 96_pin_Loopback, 25_pin_loopback, or Echo_TTY test mode. |
| Ptr=             | Enables or disables the Printer subtest. |
| B=               | Sets the baud rate to 110, 300, 600, 1200, 2400, 4800, 9600, 19200, or 38400. The default is 9600. To use the 38400 rate, only one port at a time can be tested, and the Internal test must be disabled. |
| #of_stop_bits    | Toggles the number of stop bits between 1 or 2. The default is 1. |
| character_size   | Sets character size as a number between 5 and 8. |
### TABLE 50-4  spiftest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P=parity</td>
<td>Specifies the parity as none, odd, or even. The default is none.</td>
</tr>
<tr>
<td>F=flow_control</td>
<td>Specifies flow control as xonxoff,rtscs, or both.</td>
</tr>
<tr>
<td>Data=test_pattern</td>
<td>Specifies test pattern as 0x55555555, 0xAAAAAA, or random.</td>
</tr>
<tr>
<td>sp=serial_port</td>
<td>Specifies the terminal and serial port number, such as term/3.</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
CHAPTER 51

Serial Ports Test (sptest)

sptest checks the system’s on-board serial ports \( \{zs[0,1], zsh[0,1], se[0,1], se_{hdlc}[0,1]\} \), as well as any multi-terminal interface (ALM2) boards \( \{mcp[0-3]\} \). Data is written and read in asynchronous and synchronous modes utilizing various loopback paths.

Intervention mode must be enabled to run this test.

This test is nonscalable.

sptest dynamically probes for se, zs, zsh, and se_{hdlc} ports for testing. All error messages, warning messages, and options in the test parameter options dialog box are set dynamically to the correct port names. For example, if a system’s a and b ports are su ports, and c and d are se ports, sptest will test ports c and d. The test parameter dialog box will also show the correct ports in its menu options, such as “Test Type: c to d,” etc. See FIGURE 51-1 on page 308 for an example of the test parameter dialog box.
TABLE 51-1  Serial Port Tests

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asynchronous testing</td>
<td>This mode uses the asynchronous communication protocol as described in the <code>zs(7D)</code> and <code>se(7D)</code> man pages. The <code>termio(7I)</code> interface is used for configuring port characteristics. The user can select the loopback paths to use, the amount of data to transfer, and the baud rate to use. The test writes and reads data through the loopback path and compares the data to the original data. The test first sends a single character. If no errors or timeouts are detected, the rest of the data is simultaneously written and read, then compared.</td>
</tr>
<tr>
<td>Synchronous testing</td>
<td>This mode uses the synchronous hdlc-framing interface as described in the <code>zsh(7D)</code> and <code>se_hdlc(7D)</code> man pages. Data is written and read in checksum-protected packets. The user can select the loopback paths to use and the clock rate. The synchronous test runs in three phases: 1. The first phase looks for activity on the port. If no activity is detected for at least four seconds, the test proceeds to the next phase. If activity is detected <code>sptest</code> exits with an error. 2. The second phase attempts to send and receive one packet. If no packets are detected after five attempts, the test exits with an error. If a packet is returned, the result is compared to the original. If the length and content of the packets do not match exactly, the test exits with an error. 3. The third phase attempts to send many packets through the loop. Some packet drops are to be expected especially on a heavily loaded system. The test allows a percentage of the packets to be dropped. The user can set the drop tolerance between 0 percent and 99 percent. The default is 20 percent. If the system is extremely busy then the drop tolerance should be increased. Each packet is compared with its original for length and content. If a mismatch is detected, the test exits with an error.</td>
</tr>
</tbody>
</table>
sptest Synchronous Testing Software Requirements

If you have zs(7D) serial ports on your machine, the synchronous devices may not exist. Look in the /dev directory for zsh (where h=0 and/or 1). If they do not exist, you can create them.

▼ To Create Synchronous Devices

1. Verify that the following two lines are in the /etc/devlink.tab file. If they are not there, add them.

   ```
   type=ddi_pseudo;name=zsh zsh\M0
   type=ddi_pseudo;name=clone;minor=zsh zsh
   ```

   **Caution** – The white spaces in the lines above must be a single tab character before and after the zsh variables; using spaces will not work.

2. When the lines have been added to the /etc/devlink.tab file, change directories to /kernel/drv and run the add_drv zsh command.

3. If this command does not work, run the rem_drv zsh command and then run the add_drv zsh command again.

sptest 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.
There are a variety of loopback paths available. The internal loopback paths do not require an external connector. Their availability depends on the device. The `zs(7D)` device has an internal path for synchronous mode and the `se(7D)` device has an internal path for asynchronous mode. The external loopback connectors are described in Appendix A. The exact type of loopback connector required depends on the system I/O panel.

Some examples of loopback test commands follow. For a full description of command-line syntax, see “`sptest Command-Line Syntax`” on page 51313.
To test /dev/term/b from the command line using internal loopback, type the following:

```
% ./sptest -vf -o M=async,T=b,L=I
```

To test /dev/term/b from the command line using external loopback, type:

```
% ./sptest -vf -o M=async,T=b,L=P
```

For zs(7D) machines, the internal loopback path is only active in synchronous mode:

```
% ./sptest -vf -o M=sync,T=b,L=I
```

The following table lists the possible devices for each port. Below, “a” represents port a of the CPU board (motherboard), “b” represents port b of the CPU board, and the device names of the ports for each CPU board are listed.

<table>
<thead>
<tr>
<th>CPU</th>
<th>Port</th>
<th>Async Device</th>
<th>Sync Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>a</td>
<td>zs0 or se0</td>
<td>zsh0 or se_hdlc0</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>zs1 or se1</td>
<td>zsh1 or se_hdlc1</td>
</tr>
<tr>
<td>1</td>
<td>a</td>
<td>zs2</td>
<td>zsh2 *</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>zs3</td>
<td>zsh3 *</td>
</tr>
<tr>
<td>2</td>
<td>a</td>
<td>zs4</td>
<td>zsh4 *</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>zs5</td>
<td>zsh5 *</td>
</tr>
</tbody>
</table>

* Currently, only zsh0 and zsh1 are supported by device drivers.
### TABLE 51-3  sptest Options

<table>
<thead>
<tr>
<th>sptest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Type</td>
<td>Selects how the test will run. Test options include:</td>
</tr>
<tr>
<td></td>
<td>• a= runs the test on port a</td>
</tr>
<tr>
<td></td>
<td>• b= runs the test on port b</td>
</tr>
<tr>
<td></td>
<td>• a_b= runs the test on ports a and b sequentially</td>
</tr>
<tr>
<td></td>
<td>• a_b_concurrent= runs the test on port a and port b concurrently.</td>
</tr>
<tr>
<td>Loopback Type</td>
<td>Selects the loopback test. Options include:</td>
</tr>
<tr>
<td></td>
<td>• Internal is an internal path for a, b, a_b, and a_b_concurrent test types.</td>
</tr>
<tr>
<td></td>
<td>• Plug_a_to_a_b_to_b is an external loopback plug for a, b, a_b, and a_b_concurrent test types.</td>
</tr>
<tr>
<td></td>
<td>• no_modem_a_to_b is an external loopback cable for a_to_b and a_to_b_concurrent test types.</td>
</tr>
<tr>
<td></td>
<td>• Modem_a_to_b is an external loopback cable with a modem attached to generate synchronous Transmit and Receive clocks in synchronous mode. The modem a_to_b external loopback type is intended for Sun internal use only. It requires custom equipment that is not available.</td>
</tr>
<tr>
<td>Test Mode</td>
<td>Selects the mode to put the serial device into before running the test. The modes available are Asynchronous, Synchronous or Both. When Both is selected, the test runs in Asynchronous mode then Synchronous mode.</td>
</tr>
<tr>
<td>Data Type</td>
<td>Selects the data pattern to transfer. The user can select:</td>
</tr>
<tr>
<td></td>
<td>• Random</td>
</tr>
<tr>
<td></td>
<td>• Sequential</td>
</tr>
<tr>
<td></td>
<td>• Alphanumeric</td>
</tr>
<tr>
<td></td>
<td>• 0x00-0xff</td>
</tr>
<tr>
<td>Async Baud Rate</td>
<td>Selects the baud rate for Asynchronous mode testing. The valid rates are: 110, 300, 600, 1200, 4800, 9600, 19200, 38400, 57600, 76800, 115200, 153600, 230400, 307200, 460800, and ALL. The default rate is 9600 baud. Some platforms can only support up to 38400 or 76800. The test will return an error if you try to use a higher baud rate than is supported. For baud rates greater than 153600 the serial line drivers must be set for RS-423 mode and not RS-232 mode. The RS-423 and RS-232 modes are usually selected by a hardware jumper on the motherboard. Consult your hardware installation manual for more information.</td>
</tr>
<tr>
<td>Async Data Size</td>
<td>Selects the total number of bytes to transfer in Asynchronous mode. This can range from 1 to 10000.</td>
</tr>
</tbody>
</table>
Async Flow Control Selects the type of flow control to use in asynchronous mode testing. The user can select Hardware (RTS/CTS), Software (XON/XOFF) or None. The default depends on the loopback type. Software flow control is not allowed on a, b, a_b, or a_b_concurrent loopback types.

Sync Baud Rate Selects the device generated clock rate for synchronous mode testing. The valid rates are from 110 to 230400. The rate does not have to be a specific value as required for async mode baud rates. The default rate is 9600. Some platforms can only support up to 38400 or 76800. The test will return an error if you try to use a higher rate then is supported. For rates greater than 100000 the serial line drivers must be set for RS-423 mode and not RS-232 mode. The RS-423 and RS-232 modes are usually selected by a hardware jumper on the motherboard. Consult your hardware installation manual for more information.

Sync Packet Drop Tolerance Selects the tolerance level of Synchronous mode dropped packets during the many_packets subtest. The default is 20 percent. The valid range is from 0 percent to 99 percent. Some packet drops are expected especially at higher clock rates and on a heavily loaded system.

Sync Poll Wait Selects the number of seconds in additional time to wait for a Synchronous mode packet to be sent. Additional time may be needed when there is heavy system activity and time-outs are being detected. In general, the user can decrease the value to 0 when the system load is light or increase the value when there is a heavy system load.

<table>
<thead>
<tr>
<th>sptest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Async Flow Control</td>
<td>Selects the type of flow control to use in asynchronous mode testing. The user can select Hardware (RTS/CTS), Software (XON/XOFF) or None. The default depends on the loopback type. Software flow control is not allowed on a, b, a_b, or a_b_concurrent loopback types.</td>
</tr>
<tr>
<td>Sync Baud Rate</td>
<td>Selects the device generated clock rate for synchronous mode testing. The valid rates are from 110 to 230400. The rate does not have to be a specific value as required for async mode baud rates. The default rate is 9600. Some platforms can only support up to 38400 or 76800. The test will return an error if you try to use a higher rate then is supported. For rates greater than 100000 the serial line drivers must be set for RS-423 mode and not RS-232 mode. The RS-423 and RS-232 modes are usually selected by a hardware jumper on the motherboard. Consult your hardware installation manual for more information.</td>
</tr>
<tr>
<td>Sync Packet Drop Tolerance</td>
<td>Selects the tolerance level of Synchronous mode dropped packets during the many_packets subtest. The default is 20 percent. The valid range is from 0 percent to 99 percent. Some packet drops are expected especially at higher clock rates and on a heavily loaded system.</td>
</tr>
<tr>
<td>Sync Poll Wait</td>
<td>Selects the number of seconds in additional time to wait for a Synchronous mode packet to be sent. Additional time may be needed when there is heavy system activity and time-outs are being detected. In general, the user can decrease the value to 0 when the system load is light or increase the value when there is a heavy system load.</td>
</tr>
</tbody>
</table>
sptest Test Modes

sptest supports all three SunVTS test modes.

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Attempts to open the port to determine if the device is connected. If it fails and the port is not busy, the test exits with an error. If it is successful or fails with a busy or exclusive use error, then the port is considered connected, and the test passes.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Performs the selected loopback test.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
sptest Command-Line Syntax

```
/opt/SUNWvts/bin/sptest standard_arguments -o dev=device_name, porta=port_name, T=test_type, L=loopback_type, M=mode, D=data_pattern, AB=async_baud_rate, S=async_data_size, par=none|even|odd, BS=1|10|100|1000|3000|5000|10000, F=flow_control, B=sync_baud_rate, DP=sync_drop_tolerance, P=sync_poll_wait
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| `dev=device_name` | Identifies the serial port(s) to test. There is no default value. You must specify a device name such as:  
|               |   • se0,  
|               |   • zs0, zs1  
|               |   • zs2, zs3  
|               |   • ..., ... |
| `porta=port_name` | The name of the first device of a serial device pair. The default is a. |
| `T=test_type` | Specifies the type of test to run:  
|              |   • a= runs the test on port a.  
|              |   • b= runs the test on port b.  
|              |   • a_b= runs the test on ports a and b sequentially.  
|              |   • a_b_concurrent= runs the test on port a and port b concurrently.  
|              |   • a_to_b= runs the test from port a to port b. |
| `L=loopback_type` | The type of loopback connector attached to ports:  
|                   |   • No_modem_a_to_b  
|                   |   • Internal_a_to_a__b_to_b  
|                   |   • Plug_a_to_a__b_to_b  
|                   |   • Modem_a_to_b |
| `M=mode` | The default test mode is asynchronous. Specify one of the following modes:  
|          |   • asynch  
|          |   • synch  
|          |   • both |
| `D=data_pattern` | Selects the data pattern to transfer. The user can select:  
|                 |   • Random  
|                 |   • Sequential  
|                 |   • Alphanumeric  
|                 |   • 0x00-0xFF |
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 “32-Bit  
and 64-Bit Tests” on page 13.
SunHSI Board Test (sunlink)

The sunlink test verifies the functionality of the SBus and PCI bus SunHSI™ boards by using the HDLC protocol. sunlink initializes and configures the selected channel.

Next, sunlink opens a datagram socket and tries to modify the socket to accept ioctl communications with the driver, and receive synchronous mode information from it.

sunlink then opens the ports, linking the upper and lower layers with ioctl calls. After initialization, this test checks for activity before attempting to send or receive data. An error message is returned if activity is detected; otherwise the transmit buffer is filled with random data. Random data is used by default. You may also specify other patterns. The data is then transmitted. If the transmission succeeds, sunlink then receives the returned data and verifies that it is identical to what was sent. Finally, statistics about the send and receive are gathered from the socket.

A full sunlink test takes approximately eight minutes per port and makes a brief check of the board ports before the actual test begins. If the port is bad, the test immediately aborts and returns an error message.

sunlink Test Requirements

This test will not pass unless you install the correct loopback connectors or port to port cables on the ports you are testing. The ports specified for test in the Options dialog box must have loopback connectors attached. See Appendix A for loopback connector part numbers and wiring instructions.
sunlink Options

To reach the 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.

The Configuration field displays the available ports. (See FIGURE 52-1.)

FIGURE 52-1 sunlink Test Parameter Options Dialog Box
TABLE 52-1  sunlink Options

<table>
<thead>
<tr>
<th>sunlink Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock source</td>
<td>Select either the onboard clock or an external clock for use when using sunlink. To use the external clock option, the transmit, receive, and clock data lines must be physically looped back.</td>
</tr>
<tr>
<td>Internal Loopback</td>
<td>Enables or disables internal loopback tests. Internal Loopback is only needed when the Loopback setting is not port-to-port, and the clock source is onboard.</td>
</tr>
<tr>
<td>Baud Rate</td>
<td>Specifies the bit rate transfer speed from 9600bs to 2.048mbs.</td>
</tr>
<tr>
<td>Port</td>
<td>Specifies the loopback type—simple single external port loopback, multiple external port loopback, and port-to-port external loopback.</td>
</tr>
</tbody>
</table>

sunlink Loopback Connectors

Refer to Appendix A of this manual for information on Sunlink™ loopback cables and loopback connectors. Refer to the High Speed Serial Interface hardware manuals for information on null modem cables.
sunlink Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full set of tests. (Offline)</td>
</tr>
</tbody>
</table>

sunlink Command-Line Syntax

```
/opt/SUNWvts/bin/sunlink standard_arguments -o dev=device_name, p=port#, P=data_pattern, brate=speed_n, I,C=clocksource
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the device to be tested. Use hih0 for the HDLC protocol.</td>
</tr>
<tr>
<td>p=ports</td>
<td>Specifies the port number to be tested.</td>
</tr>
<tr>
<td>P=data_pattern</td>
<td>Specifies the data_pattern as one of the following:</td>
</tr>
<tr>
<td></td>
<td>• c—Character (0x55)</td>
</tr>
<tr>
<td></td>
<td>• i—Incrementing</td>
</tr>
<tr>
<td></td>
<td>• d—Decrementing</td>
</tr>
<tr>
<td></td>
<td>• r—Random (default)</td>
</tr>
<tr>
<td>brate=speed_n</td>
<td>Specifies the bit rate transfer speed from 9600bs to 2.048mbs.</td>
</tr>
<tr>
<td>I</td>
<td>Enables internal loopback for HSI.</td>
</tr>
<tr>
<td>c=clocksource</td>
<td>Specifies the clock source value as one of the following:</td>
</tr>
<tr>
<td></td>
<td>• B—Onboard clock source</td>
</tr>
<tr>
<td></td>
<td>• E—External clock source</td>
</tr>
</tbody>
</table>

The following is a typical command-line syntax for testing a SunHSI board:

```
# /opt/SUNWvts/bin/sunlink -o dev=hih0,P=0+1+2+3,brate=100000
```
This command tests the internal loopback for ports 0, 1, 2, and 3. It does not run for the port to port internal loopback test.

**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 “32-Bit and 64-Bit Tests” on page 13.
CHAPTER 53

SunPCI II Test (sunpci2test)

The sunpci2test tests the SunPCI™ II card, which is a PC processor embedded in an add-on card. This test consists of approximately 150 POST routines that perform diagnostic, hardware detection, and initialization functions. This test issues a reset, then launches POST in the SunPCI II BIOS to check the devices. Finally, the sunpci2test runs bridge and system diagnostics tests.

sunpci2test Test Requirements

Before running the test, the X-window for Microsoft Windows must be shut down. If this is not done, the test will not launch.

▼ To Shut Down Microsoft Windows and the SunPCI II Card:

1. Click Start button in Microsoft Windows.
2. Click Shut Down.
   The shutdown window appears. Wait for the “It is now safe to shut off your PC” message.
3. Select “File” from the SunPCI window.
4. Select “Exit” from the file menu.
5. Click OK.
**sunpci2test Options**

To reach the 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.

![sunpci2test Test Parameter Options Dialog Box](image)

*FIGURE 53-1 sunpci2test Test Parameter Options Dialog Box*

`sunpci2test` only runs with the default parameters in place. Thus, this test does not allow any options to be configured specifically for an individual system. The number of instances is preset to 1 (the default value), as only one local copy of the test is supported.
sunpci2test Test Modes

TABLE 53-1  sunpci2test Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Runs the full set of tests</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests</td>
</tr>
</tbody>
</table>

sunpci2test Command-Line Syntax

/\opt/SUNWvts/bin/sunpci2test standard_arguments

Note – There are no test-specific options for sunpci2test.

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 “32-Bit and 64-Bit Tests” on page 13.
SuperI/O Test (sutest)

sutest checks the on-board system serial ports (su[0,1]). Data is written and read in asynchronous mode, using various loopback paths. You can select the loopback paths to use, the amount of data to transfer, and the baud rate.

The test writes and reads data through the loopback path and compares the data to the original data. The test first sends a single character. If no errors or timeouts are detected, the rest of the data is simultaneously written and read, then compared.

This test uses the asynchronous communication protocol. The termio(7I) interface is used for configuring port characteristics.

For CPU 0, port a on the CPU board (motherboard) uses the su0 asynchronous device, while port b uses su1.

Intervention mode must be enabled to run this test.

This test is nonscalable.

Loopback Connectors

This test requires null modem and plug connectors, which are described in Appendix A “Loopback Connectors”.

There are a variety of loopback paths available. The exact type of loopback connector required depends on the system I/O panel.

The loopback for the “Null Modem a to b” option is a female-to-female plug. Its pin configuration is the same as the one described for the “9-Pin to 9-Pin Port-to-Fort Loopback Cable” on page A378.

The loopback for the “Plug a to a” option is described in the section “9-pin Female Single-Fort Loopback Plug” on page A376.
Note – sutest supports any pair of ports, not just a and b.

sutest Options

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

![FIGURE 54-1 sutest Test Parameter Options Dialog Box](image)
### TABLE 54-1 sutest Options

<table>
<thead>
<tr>
<th>sutest Options</th>
<th>Description</th>
</tr>
</thead>
</table>
| Test Type      | Selects how the test will run. Test options include:  
  • a= runs the test on port a  
  • b= runs the test on port b  
  • a_b= runs the test on ports a and b sequentially  
  • a_b_concurrent= runs the test on port a and port b concurrently. |
| Loopback Type  | Selects the loopback test. Options include:  
  • Internal_a_to_a__b_to_b_ is an internal path for a, b, a_b, and a_b_concurrent test types.  
  • Plug_a_to_a__b_to_b is an external loopback plug for a, b, a_b, and a_b_concurrent test types.  
  • null_modem_a_to_b is an external loopback cable for a_to_b and a_to_b_concurrent test types. |
| Data Type      | Selects the data pattern to transfer. The user can select:  
  • Random  
  • Sequential  
  • Alphanumeric  
  • 0x00-0xff |
| Async Baud Rate| Selects the baud rate for Asynchronous mode testing. The valid rates are: 50, 110, 300, 600, 1200, 38400, 9600, 19200, 38400, 57600, 115200, and ALL. The default rate is 9600 baud. Some platforms can only support up to 38400 or 76800. The test will return an error if you try to use a higher baud rate than is supported. For baud rates greater than 153600 the serial line drivers must be set for RS-423 mode and not RS-232 mode. The RS-423 and RS-232 modes are usually selected by a hardware jumper on the motherboard. Consult your hardware installation manual for more information. |
| User Defined Baud Rate | Allows the user to set new baud rate values for the test. First select the User Defined option from the Async Baud Rate menu. Then enter the new value in the User Defined Baud Rate field. |
| Async Data Size | Selects the total number of bytes to transfer in Asynchronous mode. This can range from 1 to 10000. |
| Async Flow Control | Selects the type of flow control to use in asynchronous mode testing. The user can select Hardware (RTS/CTS), Software (XON/XOFF) or None. The default depends on the loopback type. Software flow control is not allowed on a, b, a_b, or a_b_concurrent loopback types. |
sutest Test Modes

sutest supports all three SunVTS test modes.

**TABLE 54-2** sutest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Tries to open the port to determine if the device is connected. If it fails and the port is not busy, the test exits with an error. If it is successful or fails with a busy or exclusive use error, then the port is considered connected, and the test passes.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Performs the selected loopback test.</td>
</tr>
</tbody>
</table>

**sutest Command-Line Syntax**

```
/opt/SUNWvts/bin/sutest standard_arguments -o dev=device_name, porta=port_name, T=test_type, L=loopback_type, D=data_pattern, AB=baud_rate, BS=write_size, F=flow_control
```

**TABLE 54-3** sutest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Identifies the serial port(s) to test. There is no default value. You must specify a device name such as su0.</td>
</tr>
<tr>
<td>porta=port_name</td>
<td>The name of the first device of a serial device pair. The default is a.</td>
</tr>
<tr>
<td>T=test_type</td>
<td>Specifies the type of test to run:</td>
</tr>
<tr>
<td></td>
<td>• a= runs the test on port a.</td>
</tr>
<tr>
<td></td>
<td>• b= runs the test on port b.</td>
</tr>
<tr>
<td></td>
<td>• a_b= runs the test on ports a and b sequentially.</td>
</tr>
<tr>
<td></td>
<td>• a_b_concurrent= runs the test on port a and port b concurrently.</td>
</tr>
<tr>
<td></td>
<td>• a_to_b= runs the test from port a to port b.</td>
</tr>
<tr>
<td>Argument</td>
<td>Explanation (Continued)</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------</td>
</tr>
</tbody>
</table>
| **L=loopback_type** | The type of loopback connector attached to ports:  
• Internal_a_to_a__b_to_b  
• Null_modem_a_to_b  
• Plug_a_to_a__b_to_b |
| **D=data_pattern** | Selects the data pattern to transfer. The user can select:  
• Random  
• Sequential  
• Alphanumeric  
• 0x00-0xFF |
| **AB=baud_rate** | Asynchronous baud rate (default = 9600). The valid values are between 110 - 460800.  
Note: Some platforms can only support asynchronous baud rates up to 38400 or 76800. For baud rates greater than 153600 the serial line drivers must be set for RS-423 mode and not RS-232 mode. |
| **BS=write_size** | Asynchronous mode write size; from 1 to 10000 bytes. |
| **F=flow_control** | Asynchronous mode flow control  
• Hardware (RTS/CTS)  
• Software (xon/xoff)  
• None |

**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 “32-Bit and 64-Bit Tests” on page 13.
System Test (systest)

systest checks the CPU board by exercising the I/O, memory, and CPU channels simultaneously as threads. There is no quick test option for systest; it is a CPU stress test.

systest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User's Guide for more details.
FIGURE 55-1  systest Test Parameter Options Dialog Box
systest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full set of tests. (Offline)</td>
</tr>
</tbody>
</table>

systest Command-Line Syntax

```
/opt/SUNWvts/bin/systest standard_args -o -io=Enable|Disable -mem=Enable|Disable, -cpu=Enable|Disable, -dev=system
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>io</td>
<td>Enables or Disables the Asynch I/O Test</td>
</tr>
<tr>
<td>mem</td>
<td>Enables or Disables the Memory Test</td>
</tr>
<tr>
<td>cpu</td>
<td>Enables or Disables the Cpu Test</td>
</tr>
<tr>
<td>dev</td>
<td>Specifies the pseudo device name</td>
</tr>
</tbody>
</table>

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 “32-Bit and 64-Bit Tests” on page 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 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.
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 56-1.

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 2.6 and Solaris 7-9 operating environments and compatible releases.

---

FIGURE 56-1 tapetest Test Parameter Options Dialog Box (4-mm Tape Drive)
**Note** – This test does not immediately stop after being disabled.

**Note** – Selecting non-default 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 56-1.

<table>
<thead>
<tr>
<th>TABLE 56-1 tapetest Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tapetest Options</strong></td>
</tr>
<tr>
<td><strong>Type</strong></td>
</tr>
<tr>
<td><strong># of Tapes</strong></td>
</tr>
</tbody>
</table>
| **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. |
### TABLE 56-1 tapetest Options

<table>
<thead>
<tr>
<th>tapetest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Length</strong></td>
<td>The amount of the tape to be tested. The choices are:</td>
</tr>
<tr>
<td></td>
<td>• <strong>EOT</strong>: The default; tests to the entire tape.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Long</strong>: The SCSI tape tests 70,000 blocks of the tape.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Short</strong>: Only the first 1000 blocks are tested.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Specified</strong>: You must type the number of blocks to be tested in the <code># of blocks</code> field.</td>
</tr>
<tr>
<td><strong># of Blocks</strong></td>
<td>If you select Specified under the Length option, you must type the number of blocks you want to test.</td>
</tr>
<tr>
<td><strong>Blocksize</strong></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Note1: This option is only available in command line interface mode.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td><strong>File Test</strong></td>
<td>The tape file test sequence is as follows:</td>
</tr>
<tr>
<td></td>
<td>1. Writes three files.</td>
</tr>
<tr>
<td></td>
<td>2. Rewinds.</td>
</tr>
<tr>
<td></td>
<td>3. Reads part of the first file.</td>
</tr>
<tr>
<td></td>
<td>4. Forward spaces to the start of the second file.</td>
</tr>
<tr>
<td></td>
<td>5. Reads the second file.</td>
</tr>
<tr>
<td></td>
<td>6. Forward spaces to the start of the third file.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Retension</strong></td>
<td>When enable is selected, the program retensions the tape.</td>
</tr>
<tr>
<td><strong>Media Test Method</strong></td>
<td>• <strong>Sync I/O</strong>—tapetest reads and or writes the number of blocks selected in Length.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Async I/O</strong>—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.</td>
</tr>
<tr>
<td></td>
<td>Note – When testing Tandberg QIC drives, Async I/O testing is restricted to read-only due to asynchronous behavior differences with other tape drives.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>
**tapetest Test Modes**

The `tapetest` supports all three modes. It performs different test schemes on the tape device, according to the mode you select.

### TABLE 56-2 tapetest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td><code>tapetest</code> 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 <code>tapetest</code> fails if the open operation is unsuccessful for any reason other than the drive is busy.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td><code>tapetest</code> checks the status, rewinds the tape, erases and retensions it. If the device is a cartridge tape, <code>tapetest</code> writes a pattern to <code>nblks</code> or <code>eot</code> (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.</td>
</tr>
</tbody>
</table>

---

**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 56-3 tapetest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the <code>device_name</code> of the tape drive (required).</td>
</tr>
<tr>
<td>s=block_count</td>
<td>Specifies the number of blocks to be tested.</td>
</tr>
<tr>
<td>d=density</td>
<td>Specifies the density of the tape to open.</td>
</tr>
<tr>
<td>m=mode</td>
<td>Enables either the Write_Read or Read_Only tests.</td>
</tr>
<tr>
<td>l=length</td>
<td>Specifies the length of the test (EOT, Specified, Long, or Short).</td>
</tr>
</tbody>
</table>
### TABLE 56-3  tapeTest Command-Line Syntax (Continued)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation (Continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td>method=method</td>
<td>Specifies the media test method (SyncI/O and or AsyncI/O) used.</td>
</tr>
<tr>
<td></td>
<td>Note: This option does not invoke the SCSI message “synchronous data transfer request. It is only asynchronous or synchronous in nature.</td>
</tr>
<tr>
<td>ft=enable</td>
<td>disables</td>
</tr>
<tr>
<td>ret=enable</td>
<td>disables</td>
</tr>
<tr>
<td>dat=dat_type</td>
<td>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.</td>
</tr>
<tr>
<td>8mm=8mm_type</td>
<td>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.</td>
</tr>
<tr>
<td>num=magazine_size</td>
<td>If you are testing a tape library, specify the magazine size.</td>
</tr>
<tr>
<td>blocksize=block_size</td>
<td>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</td>
</tr>
</tbody>
</table>

**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 “32-Bit and 64-Bit Tests” on page 13.
S24 Frame Buffer Test (tcxtest)

Through a series of protocol, memory, acceleration, and colormap tests, tcxtest checks the functionality of the S24 Frame Buffer SBus card used on the SPARCstation 5 and checks the FSV (fast SBus video) ASIC on the SPARCstation 4 motherboard.

Note – Disable all screen savers before testing any graphics device. Type xset s off at a UNIX prompt to disable the Solaris screen saver.

For full instructions on testing frame buffers, see “Testing Frame Buffers” on page 18.

tcxtest Test Groups

tcxtest has four distinct test groups.

AFX Protocol tests (in 8/16/32/64-bit mode):
- WRC

Frame Buffer Memory tests (in 8/16/32/64-bit14 mode):
- address
- constant
- random

Acceleration tests (both User and Raw modes):
- bit
- stip

Colormap and Cursor tests:
- cursor (does not apply to SPARCstation4)
- colormap
### tcxtest Subtests

**TABLE 57-1 tcxtest Subtests**

<table>
<thead>
<tr>
<th>tcxtest Subtests</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WRC</strong></td>
<td>By performing multiple writes and reads, and then verifying the results, the WRC test exercises the FIFO inside the S24 chip. The WRC test is composed of these three subtests: <code>test_afx_alt_wr</code>, <code>test_memafx</code>, and <code>test_afx_random</code>. If these tests fail, they print an error message showing the expected and observed data.</td>
</tr>
<tr>
<td><strong>Test_afx_alt_wr</strong></td>
<td>This test performs 16 writes to alternative pages, for example, WR (Page1), WR (Page2), WR (Page1+off), WR (Page2+off), and so on. It then reads back the data and compares it with the expected results. This test also writes to the frame buffer space 16 times, followed by a write to a different page in the frame buffer space. The test then reads this data back and verifies it with the expected results.</td>
</tr>
<tr>
<td><strong>Test_memafx</strong></td>
<td>The CPU in the SWIFT chip has closely coupled interfaces for the DRAM and the AFX bus. This test checks the arbitration between the two accesses. This test performs a number of alternating writes to the AFX and the CPU memory. After writing to different locations, the test reads and verifies the data. By performing an access across the page boundaries, the test covers both the cached and non-cached accesses.</td>
</tr>
<tr>
<td><strong>Test_afx_random</strong></td>
<td>After writing to one page in the DRAM memory, the test performs a few random writes/reads to random locations in the AFX space. The test then writes to a different page in the DRAM space, where it performs random accesses. This test does not perform any data verification, it just checks to see if any of these random accesses caused a time out.</td>
</tr>
<tr>
<td><strong>Constant</strong></td>
<td>This test writes a data pattern to the whole memory. This pattern is read back and compared with the expected data. Once the memory fill operation is completed, the test reads the memory back and verifies that the value read is correct.</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td>This test writes a data pattern (which is same as the value of the address) to the whole memory. This pattern is then read back to verify that it is the correct value.</td>
</tr>
</tbody>
</table>
### tcxtest Subtests

<table>
<thead>
<tr>
<th>tcxtest Subtests</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>This test writes a random data pattern to the whole memory. This pattern is read back and compared with the expected data. After the memory fill operation is completed, the test reads the memory and verifies the values read are correct.</td>
</tr>
<tr>
<td>Blit</td>
<td>This test has two parts; the raw blit test and the user blit test. The raw blit test draws a 64x64x24 pixel image at the top-left corner of screen. Next it blits the image to the screen. The destination images are read back and compared with the original image to verify the raw blit operation has run correctly. The user blit test draws a 64x64x24 pixel image at the top left corner of screen. It then blits the image to the screen. The destination images are read back and compared with the original image. The user blit test is the same as the raw blit test, except the user blit test uses the user data space for the blit command.</td>
</tr>
</tbody>
</table>
| Stip             | This test performs numerous corner cases for stipple. The test writes to the destination with different data values using a stipple operation. The destination data is read back and verified. For the fast SBus video (SV), the following is checked:  
  - Walks 1 through pixel mask.  
  - Walks 1 through ROP bits.  
  - Walks 1 through destination byte.  
  - Walks 1 through IDX byte. |
| Cursor (does not apply to SPARCstation 4) | This test performs a data register regression test. It writes a walking 1 pattern to the cursor data registers. The data is then read back and verified with the expected results. The test is repeated using a walking 0 as the data pattern. |
| Colormap         | Loads all locations in the colormap with varying values of RGB.  
  Note: If the system being tested has a monochrome or greyscale monitor, visual color problems are undetectable. |

### tcxtest 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.
FB locking is the only test parameter option for this test. See “Testing Frame Buffers” on page 18 for details.

1. Click enable or disable to configure frame buffer locking, etc.

![FIGURE 57-1 txttest Test Parameter Options Dialog Box](image-url)
tcxtest Test Modes

TABLE 57-2 tcxtest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the appropriate subtests for the hardware platform being tested.</td>
</tr>
</tbody>
</table>

tcxtest Command-Line Syntax

```
/opt/SUNWvts/bin/tcxtest standard_arguments -o dev=device_name,lock=
E(nable)|D(isable), x=bit_mode,T=test,S=[dfb8, dfb24, dfb32]
```

TABLE 57-3 tcxtest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>dev=device_name</td>
<td>Specifies the filename of the device to be tested, for example, dev=tcx0.</td>
</tr>
<tr>
<td>lock=E(nable)</td>
<td>D(isable)</td>
</tr>
</tbody>
</table>
## TABLE 57-3  
tcxtest Command-Line Syntax (Continued)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| X=bit_mode | Specifies the data transfer size. Supported values are:  
  • 8byte  
  • 16short  
  • 32long  
  • 64double word |
| T=test | Specifies a particular test. To specify an individual test, replace test with:  
  • a=Address  
  • c=Constant  
  • r=Random  
  • b=Blit  
  • s=Stipple  
  • h=Cursor  
  • w=WRC |

Note: When you select either the Blit or Stipple test, both the User and Raw mode tests are run.

| S=[dfb8, dfb24, dfb32] | Specifies which frame buffer memory space to use.  
  • -dfb8—Dumb frame buffer 8-bit space. Memory is accessed only by bytes.  
  • -dfb24—Dumb frame buffer 24-bit space. Memory is accessed only by 24-bit reads and writes.  
  • -dfb32—Dumb frame buffer 8-bit space. Memory is accessed by 8-bit reads and writes. |

**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 “32-Bit and 64-Bit Tests” on page 13.
Sun USB Keyboard Test
(usbkbtest)

usbkbtest verifies whether the keyboard(s) attached to the USB bus are USB compliant. The test will flash the LEDs of a compliant keyboard.

usbkbtest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the SunVTS User’s Guide for more details.
FIGURE 58-1  *usbkbtest* Test Parameter Options Dialog Box
**usbkbtest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td>Runs the full test.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td>Runs the full test.</td>
</tr>
<tr>
<td>(Offline)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**usbkbtest Command-Line Syntax**

```
/opt/SUNWvts/bin/usbkbtest standard_arguments
-o dev=kbd|usb/hidn
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`dev=kbd</td>
<td>usb/hidn`</td>
</tr>
</tbody>
</table>
Virtual Memory Test (*vmemtest*)

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

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

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

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

---

*vmemtest* Swap Space Requirements

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

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

FIGURE 59-1 vmemtest Test Parameter Options Dialog Box
### TABLE 59-1  vmemtest Options

<table>
<thead>
<tr>
<th>vmemtest Options</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Mode**         | Two modes are available:  
|                  | • Regular mode tests the amount of memory and is limited by the amount of physical memory available to the system under test.  
|                  | • Page mode tests assign virtual memory one page at a time. Each page is mapped to the temporary file `/tmp/vmem.page` and is then paged out to storage once test data is written. Next, the temporary page is paged back into memory for a read and compare.  
|                  | When the Stress Test Execution option is selected, the vmemtest allocates the entire assigned system memory (through `valloc`), writing from beginning to end. The memory is then read back and compared with the original pattern, one long word at a time. |
| **Reserve**      | The Reserve option specifies the amount of memory to reserve from being tested by vmemtest. The reserved space is used for other processes running concurrently with the or SunVTS tests. The Reserve option can be used to reserve memory in addition to the default. This option applies only to a specific instance. Trying to reserve more memory than what is assigned to be tested by this instance will cause the test to fail. |
| **Test Amount**  | An amount can be specified to test the virtual memory, instead of the default. Specifying a number greater than the available memory, can cause vmemtest or other running tests to fail. |
| **vmemtest**     | The amount of memory listed in the Configuration field is equivalent to the sum of the used and available swap space amounts returned by the `swap -s` command. It indicates the amount of virtual memory found, rounded up to the nearest Kbyte. |
| **Configuration**| Specifies the number of memory errors that occur on successive memory locations before testing stops. |
Test Method
- The default selection is the Sequential test. The whole memory is tested from the beginning address to the end address in a sequence.
- Address Random test: Randomly selects memory addresses to test.
- Page Striding test: Non-contiguous memory test, implemented sequentially and non-sequentially.
  - Sequential test: tests from the first page to the last page, within a specified test range. Only one word is tested per page.
  - Non-sequential test: tests randomly from first to last page, within a specified memory range. Goes back and forth testing one word per page until all pages are tested.
- Block Copy test: Writes and reads data between two memory blocks. Each memory block is half the memory to be tested.
- File Caching test: Aimed at improving performance through the use of file caching in the Solaris kernel. This test is useful for large memory configurations. This test takes 30 to 70% less time than the Sequential test method.

Predefined Pattern
Select one of the following patterns to use for the test:
- Address—uses the virtual addresses of the tested memory locations.
- walk_1—uses a pattern that starts with 0x80000000 through 0x00000001
- walk_2—uses a pattern that starts with 0x7f000000 through 0x07000700
- 0x00000000—uses all ones and zeros for testing
- 0x5aa5aa5—uses 0x5aa5aa5 and 0xa55aa55a patterns
- 0xdb6db6db—uses 0xdb6db6db and 0x24924924 patterns
- Checkerboard—uses 0x55555555 and 0xaaaaaaa patterns.
- UserDefined—uses the pattern that is specified in the User Defined Pattern area (see below).

User Defined Pattern
Only used if the Predefined Pattern is set to UserDefined. The pattern specified should be in the form of an 8-digit, hexadecimal number such as 0x2a341234.

Instance
Specifies how many copies of the vmemtest test to run.

<table>
<thead>
<tr>
<th>TABLE 59-1</th>
<th>vmemtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Method</td>
<td></td>
<td>• The default selection is the Sequential test.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The whole memory is tested from the beginning</td>
</tr>
<tr>
<td></td>
<td></td>
<td>address to the end address in a sequence.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Address Random test: Randomly selects memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>addresses to test.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Page Striding test: Non-contiguous memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>test, implemented sequentially and non-sequentially.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Sequential test: tests from the first page to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the last page, within a specified test range.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Non-sequential test: tests randomly from first</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to last page, within a specified memory range.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Block Copy test: Writes and reads data between</td>
</tr>
<tr>
<td></td>
<td></td>
<td>two memory blocks. Each memory block is half the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>memory to be tested.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• File Caching test: Aimed at improving performance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>through the use of file caching in the Solaris</td>
</tr>
<tr>
<td></td>
<td></td>
<td>kernel. This test is useful for large memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>configurations. This test takes 30 to 70% less</td>
</tr>
<tr>
<td></td>
<td></td>
<td>time than the Sequential test method.</td>
</tr>
<tr>
<td>Predefined Pattern</td>
<td>Select one of the following patterns to use for the test:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Address—uses the virtual addresses of the tested</td>
</tr>
<tr>
<td></td>
<td></td>
<td>memory locations.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• walk_1—uses a pattern that starts with 0x8000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 through 0x00000001.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• walk_2—uses a pattern that starts with 0x7f0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 through 0x07000700.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x00000000—uses all ones and zeros for testing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x5aa5aa5—uses 0x5aa5aa5 and 0xa55aa55a patterns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0xdb6db6db—uses 0xdb6db6db and 0x24924924</td>
</tr>
<tr>
<td></td>
<td></td>
<td>patterns.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Checkerboard—uses 0x55555555 and 0xaaaaaaa</td>
</tr>
<tr>
<td></td>
<td></td>
<td>patterns.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• UserDefined—uses the pattern that is specified</td>
</tr>
<tr>
<td></td>
<td></td>
<td>in the User Defined Pattern area (see below).</td>
</tr>
<tr>
<td>User Defined Pattern</td>
<td>Only used if the Predefined Pattern is set to UserDefined. The pattern specified should be in the form of an 8-digit, hexadecimal number such as 0x2a341234.</td>
<td></td>
</tr>
<tr>
<td>Instance</td>
<td>Specifies how many copies of the vmemtest test to run.</td>
<td></td>
</tr>
</tbody>
</table>
vmemtest Test Modes

TABLE 59-2 vmemtest Test Modes

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>No</td>
<td>Not supported.</td>
</tr>
<tr>
<td>Functional (Offline)</td>
<td>Yes</td>
<td>Runs the full set of tests.</td>
</tr>
</tbody>
</table>

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

vmemtest Command-Line Syntax

`/opt/SUNWvts/bin/vmemtest standard_arguments -o mode=type, reserve=n, amount=n, bdinfo=n, cerr=number, type=n, pp=pattern, up=hex_address`

TABLE 59-3 vmemtest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode=Page</td>
<td>Regular</td>
</tr>
<tr>
<td></td>
<td>• Page—tells the write/read memory test to proceed one system memory page at a time.</td>
</tr>
<tr>
<td></td>
<td>• Regular—uses the valloc option to allocate the entire assigned memory, which is read and compared one long word at a time.</td>
</tr>
<tr>
<td>reserve=n</td>
<td>Specifies the amount of MB of virtual memory to reserve in addition to the default amount.</td>
</tr>
<tr>
<td>amount=n</td>
<td>Specifies the number of MB of memory to be tested instead of the default (maximum) amount.</td>
</tr>
<tr>
<td>bdinfo=n</td>
<td>Provides the board number information for all the CPU/memory boards in the system. For example, if board 0 and board 5 have memory, then the bdinfo=33 (2<strong>5+2</strong>0).</td>
</tr>
</tbody>
</table>
TABLE 59-3  vmemtest Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>(Continued)Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cerr</code></td>
<td>Specifies the maximum number of contiguous errors to be</td>
</tr>
<tr>
<td></td>
<td>dumped when a memory error occurs.</td>
</tr>
<tr>
<td><code>type</code></td>
<td>seq—Uses Sequential test method</td>
</tr>
<tr>
<td></td>
<td>rand—Uses Random Address test method</td>
</tr>
<tr>
<td></td>
<td>page—Uses Page Striding test method</td>
</tr>
<tr>
<td></td>
<td>block—Uses Block Copy method</td>
</tr>
<tr>
<td></td>
<td>file—Uses File Caching test method</td>
</tr>
<tr>
<td><code>pp</code></td>
<td>Specifies the test pattern to use for testing. Valid pattern</td>
</tr>
<tr>
<td></td>
<td>entries are:</td>
</tr>
<tr>
<td></td>
<td>• Address—uses the virtual addresses of the tested memory</td>
</tr>
<tr>
<td></td>
<td>locations.</td>
</tr>
<tr>
<td></td>
<td>• walk_1—uses a pattern that starts with 0x80000000 through</td>
</tr>
<tr>
<td></td>
<td>0x00000001.</td>
</tr>
<tr>
<td></td>
<td>• walk_0—uses a pattern that starts with 0x7fff7fff through</td>
</tr>
<tr>
<td></td>
<td>0x7fff7ffe.</td>
</tr>
<tr>
<td></td>
<td>• 0x00000000—uses all ones and zeros for testing.</td>
</tr>
<tr>
<td></td>
<td>• 0x5aa5aa5—uses 0x5aa5aa5 pattern.</td>
</tr>
<tr>
<td></td>
<td>• 0xdb6db6db—uses 0xdb6db6db pattern.</td>
</tr>
<tr>
<td></td>
<td>• Checkerboard—uses 0x55555555 and 0xaaaaaaaa patterns.</td>
</tr>
<tr>
<td></td>
<td>• Random—uses a random data pattern.</td>
</tr>
<tr>
<td></td>
<td>• UserDefined—uses the pattern that is specified in the</td>
</tr>
<tr>
<td></td>
<td>User Defined Pattern area (see below).</td>
</tr>
<tr>
<td><code>hex_address</code></td>
<td>Only used if the <code>pp</code> argument is set to UserDefined. The</td>
</tr>
<tr>
<td></td>
<td>pattern specified should be in the form of a 8-digit,</td>
</tr>
<tr>
<td></td>
<td>hexadecimal number such as 0x2a341234.</td>
</tr>
</tbody>
</table>

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 “32-Bit and 64-Bit Tests” on page 13.
Sun Fire Link Interconnect Test  
(*wrsmttest*)

The *wrsmttest* verifies the functionality of the Sun Fire™ Link Interconnect by checking the cluster networking hardware.

**Note** – For this test to be meaningful, the cluster must be configured before the test is run.

For details on how to configure the cluster before running *wrsmttest*, refer to the cluster network administration documentation that is shipped with the product.

The *wrsmttest* uses the Internet Control Message Protocol (ICMP) that is based on the Data Link Protocol Interface (DLPI) to test the connections between cluster nodes.

First, *wrsmttest* determines the target cluster nodes to use for testing. You can specify the target hosts in the *wrsmttest* Test Parameter menu, or if no targets are specified, *wrsmttest* sends an ICMP broadcast from the private cluster network to find them. If it fails to find the necessary targets, it performs an RPC broadcast to the RPC port mapper daemon.

After finding the cluster nodes (targets), *wrsmttest* performs the following subtests:

- Random test—sends out 256 packets with random data length and random data.
- Incremental test—sends out packets with length from minimum to maximum packet size using incremental data.
- Pattern test—sends 256 packets of maximum length, where each packet contains one test pattern where all byte patterns (0 to 0xFF) are used.

**Note** – *wrsmttest* is a scalable test. The maximum number of instances is two per WIB card.
Note – The \texttt{wrsmtest} is only supported in a 64-bit operating environment.

\section*{wrsmtest 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 \textit{SunVTS User's Guide} for more details.
FIGURE 60-1 wrsmtest Test Parameter Options Dialog Box
### TABLE 60-1  

<table>
<thead>
<tr>
<th>wrsmtest Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target Host</strong></td>
<td>Specifies one or more cluster node targets for testing. Target host entries can be either a host name or an Internet address for the <em>wrsmd</em> interface, for example, target1+target2+target3. When no target host is specified, the test finds the necessary targets through broadcasting. The default setting leaves this field empty.</td>
</tr>
</tbody>
</table>
| **Test Type**    | Specifies which subtests to run:  
  - Random test—sends out 256 packets with random data length and random data.  
  - Incremental test—sends out packets with length from minimum to maximum packet size using incremental data.  
  - Pattern test—sends 256 packets of maximum length, where each packet contains one test pattern where all byte patterns (0 to 0xFF) are used.  
  All subtests are selected by default. |
| **Number of Packets** | Specifies the number of packets to use for testing. The default is 256. |
| **Receive Timeout**  | Specifies the receive timeout value in seconds. Specify a number between 0–600 seconds. The default is 120 seconds. |
| **Number of Retries** | Sets the number of retries before an error is flagged. Specify a number between 0–128. The default is 3. |
| **Print Warning**    | Choose Enable to see warning errors, such as *retry on timeout* errors. Disabled by default. |
| **Link Max Error Limit** | Upper threshold limit for peak link errors per second for a test to pass. If the test reports peak link errors that exceed this limit, the device being tested will fail and report an error message. |
| **Link Average Error Limit** | Threshold limit for average link errors per hour for a test to pass. If the test reports average link errors that exceed this limit, the device being tested will fail and report an error message. The default is 40 errors per hour. |
**wrsmtest Test Modes**

Connection, and Functional modes are supported by `wrsmtest`. Different test schemes are performed on the cluster interconnect device based on the test mode you select.

**TABLE 60-2 wrsmtest Test Modes**

<table>
<thead>
<tr>
<th>Test Mode</th>
<th>Supported?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>Yes</td>
<td><code>wrsmtest</code> checks if the device is connected. It searches through DLPI <code>wrsmd</code> interfaces for the specified device name. If <code>wrsmtest</code> finds the device not connected, the test fails, otherwise, it returns the following message: <code>device is connected</code>.</td>
</tr>
<tr>
<td>Functional</td>
<td>Yes</td>
<td><code>wrsmtest</code> performs all three subtests (Random test, Incremental test, and Pattern test) sequentially. It allows you to specify options in such a way that <code>wrsmtest</code> performs a very stressful test. (Offline)</td>
</tr>
</tbody>
</table>

**wrsmtest Command-Line Syntax**

```
/opt/SUNWvts/bin/wrsmtest standard_arguments -o dev=interface, test=type, packets=n, pattern=hex, delay=seconds, timeout=seconds, retry=n, warn=E|D, maxerr=n, avgerr=n
```

**TABLE 60-3 wrsmtest Command-Line Syntax**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dev=interface</code></td>
<td>Identifies the cluster network interface name. The default value is <code>wrsmd0</code> for DLPI cluster networks.</td>
</tr>
<tr>
<td><code>test=type</code></td>
<td>Specifies which subtests to run. Specify <code>random</code>, <code>increment</code>, or <code>pattern</code>. Use a + symbol to list multiple subtests. The default value is <code>random+increment+pattern</code>.</td>
</tr>
<tr>
<td><code>packets=n</code></td>
<td>Specifies the number of random/pattern packets. The default is 256.</td>
</tr>
<tr>
<td><code>pattern=hex</code></td>
<td>Specifies a data pattern in hexadecimal form. The default is all patterns from 0 to 0xff.</td>
</tr>
</tbody>
</table>
TABLE 60-3  \texttt{wrsmtst} Command-Line Syntax

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{delay=seconds}</td>
<td>Indicates the time between subtests in seconds. The default is 30 seconds.</td>
</tr>
<tr>
<td>\texttt{timeout=seconds}</td>
<td>Indicates the number of seconds to wait before a timeout. The default is 1 second.</td>
</tr>
<tr>
<td>\texttt{retry=n}</td>
<td>Indicates the number of test timeout retries. The default is three retries.</td>
</tr>
<tr>
<td>\texttt{warn=E</td>
<td>D}</td>
</tr>
<tr>
<td>\texttt{maxerr=n}</td>
<td>Upper threshold limit for peak link errors per second for a test to pass. If the test reports peak link errors that exceed this limit, the device being tested will fail and report an error message.</td>
</tr>
<tr>
<td>\texttt{avgerr=n}</td>
<td>Threshold limit for average link errors per hour for a test to pass. If the test reports average link errors that exceed this limit, the device being tested will fail and report an error message. The default is 40 errors per hour.</td>
</tr>
</tbody>
</table>

\textbf{Note} – 64-bit tests are located in the \texttt{sparcv9} subdirectory: 
\texttt{/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 “32-Bit and 64-Bit Tests” on page 13.
APPENDIX A

Loopback Connectors

Loopback connectors are designed for the testing of communication ports. They take the form of either a single plug or a port-to-port cable with some communication connections shorted (looped-back).

**Note** – Loopback connectors must be wired properly and connected firmly for the Serial Port tests to work correctly. Miswired, poorly soldered, or missing loopback connectors can cause erroneous diagnostic error messages.

Table A-1 depicts the pin assignments for most loopback plugs and cables that may be used when testing a system.

**TABLE A-1** Loopback Connector Pin Assignments

<table>
<thead>
<tr>
<th>Signal Description</th>
<th>EIA #</th>
<th>CCITT #</th>
<th>RS-449 “A”</th>
<th>RS-449 “B”</th>
<th>DIN 8 8-pin round</th>
<th>DB9 9-pin</th>
<th>DB25 25-pin</th>
<th>Direction</th>
<th>Alpha ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis/Fram e Ground</td>
<td>AA</td>
<td>101</td>
<td>1</td>
<td>NC*</td>
<td>NC*</td>
<td>1</td>
<td>None</td>
<td>AA</td>
<td></td>
</tr>
<tr>
<td>Transmit Data (TxDa)</td>
<td>BA</td>
<td>103</td>
<td>4</td>
<td>22</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>Output</td>
<td>BA</td>
</tr>
<tr>
<td>Receive Data (RxDa)</td>
<td>BB</td>
<td>104</td>
<td>6</td>
<td>24</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>Input</td>
<td>BB</td>
</tr>
<tr>
<td>Request To Send (RTSa)</td>
<td>CA</td>
<td>105</td>
<td>7</td>
<td>25</td>
<td>6</td>
<td>7</td>
<td>4</td>
<td>Output</td>
<td>CA</td>
</tr>
<tr>
<td>Clear To Send (CTSa)</td>
<td>CB</td>
<td>106</td>
<td>9</td>
<td>27</td>
<td>2</td>
<td>8</td>
<td>5</td>
<td>Input</td>
<td>CB</td>
</tr>
<tr>
<td>Data Set Ready (DSRa)</td>
<td>CC</td>
<td>107</td>
<td>11</td>
<td>29</td>
<td>NC*</td>
<td>6</td>
<td>6</td>
<td>Input/output</td>
<td>CC</td>
</tr>
</tbody>
</table>
TABLE A-1  Loopback Connector Pin Assignments (Continued)

<table>
<thead>
<tr>
<th>Signal Description</th>
<th>EIA</th>
<th>CCITT #</th>
<th>RS-449 “A”</th>
<th>RS-449 “B”</th>
<th>DIN 8 8-pin round</th>
<th>DB9 9-pin</th>
<th>DB25 25-pin</th>
<th>Direction</th>
<th>Alpha ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Ground</td>
<td>AB</td>
<td>102</td>
<td>19</td>
<td>NC*</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>None</td>
<td>AB</td>
</tr>
<tr>
<td>Data Carrier Detect</td>
<td>CF</td>
<td>109</td>
<td>13</td>
<td>31</td>
<td>7</td>
<td>1</td>
<td>8</td>
<td>Input</td>
<td>CF</td>
</tr>
<tr>
<td>Transmit Clock In</td>
<td>DB</td>
<td>114</td>
<td>5</td>
<td>23</td>
<td>NC*</td>
<td>NC*</td>
<td>15</td>
<td>Input</td>
<td>DB</td>
</tr>
<tr>
<td>Receive Clock In</td>
<td>DD</td>
<td>115</td>
<td>8</td>
<td>26</td>
<td>8</td>
<td>NC*</td>
<td>17</td>
<td>Input</td>
<td>DD</td>
</tr>
<tr>
<td>Data Terminal Ready</td>
<td>CD</td>
<td>108</td>
<td>12</td>
<td>30</td>
<td>1</td>
<td>4</td>
<td>20</td>
<td>Output</td>
<td>CD</td>
</tr>
<tr>
<td>External Clock Out</td>
<td>DA</td>
<td>113</td>
<td>17</td>
<td>35</td>
<td>NC*</td>
<td>NC*</td>
<td>24</td>
<td>Output</td>
<td>DA</td>
</tr>
<tr>
<td>Secondary Data Carrier Detect</td>
<td>SC</td>
<td>122</td>
<td>NC*</td>
<td>NC*</td>
<td>NC*</td>
<td>NC*</td>
<td>12</td>
<td>Input</td>
<td>SCF</td>
</tr>
<tr>
<td>Secondary Clear to Send</td>
<td>SC</td>
<td>121</td>
<td>NC*</td>
<td>NC*</td>
<td>NC*</td>
<td>NC*</td>
<td>13</td>
<td>Input</td>
<td>SCB</td>
</tr>
<tr>
<td>Secondary Transmit Data</td>
<td>SB</td>
<td>118</td>
<td>NC*</td>
<td>NC*</td>
<td>NC*</td>
<td>NC*</td>
<td>14</td>
<td>Output</td>
<td>SBA</td>
</tr>
<tr>
<td>Secondary Receive Data</td>
<td>SB</td>
<td>119</td>
<td>NC*</td>
<td>NC*</td>
<td>NC*</td>
<td>NC*</td>
<td>16</td>
<td>Input</td>
<td>SBB</td>
</tr>
<tr>
<td>Secondary Request to Send</td>
<td>SC</td>
<td>120</td>
<td>NC*</td>
<td>NC*</td>
<td>NC*</td>
<td>NC*</td>
<td>19</td>
<td>Output</td>
<td>SCA</td>
</tr>
</tbody>
</table>

*NC = No connection
25-Pin RS-232 Loopback Plug

The RS-232 and RS-423 single-port loopback plug is a specially wired male DB-25 connector. It is plugged in to a serial port in the back of the system under test.

**FIGURE A-1** 25-pin RS-232 Loopback Plug Wiring Diagram

<table>
<thead>
<tr>
<th>Connect:</th>
<th>First connector</th>
<th>Second connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin 3</td>
<td>Pin 2</td>
<td></td>
</tr>
<tr>
<td>Pins 6 and 8</td>
<td>Pin 20</td>
<td></td>
</tr>
<tr>
<td>Pins 15 and 17</td>
<td>Pin 24</td>
<td></td>
</tr>
</tbody>
</table>

25-pin RS-232 Port-to-Port Loopback Cable

Use these wiring instructions for 25-pin RS-232 and RS-423 port to 25-pin RS 232 and RS 423 port loopback cables (two DB-25 connections). It is plugged into a pair of serial ports in the back of the system under test. Both connectors are male.
8-Pin to 8-Pin Loopback Cable

Use these wiring directions for 8-pin round DIN RS-232 port to RS-423 to 8-pin round-DIN RS-232 and RS-423 port loopback cable. Both connectors are male.
8-Pin Loopback Plug

Use these wiring directions for male 8-pin round-DIN RS-232 and RS-423 single-port loopback plugs.

Pin 8, Receive clock In (DD), remains unconnected.
25-pin Port A-to-Port B Loopback Plug

Use these wiring directions for a 25-pin Port A to Port B loopback plug for most systems.

<table>
<thead>
<tr>
<th>Connect: First connector to Second connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin 16</td>
</tr>
<tr>
<td>Pin 3</td>
</tr>
<tr>
<td>Pin 13</td>
</tr>
<tr>
<td>Pin 5</td>
</tr>
<tr>
<td>Pins 6 and 8</td>
</tr>
<tr>
<td>Pin 12</td>
</tr>
<tr>
<td>Pin 18</td>
</tr>
<tr>
<td>Pins 15 and 17</td>
</tr>
</tbody>
</table>

FIGURE A-5  Port A-to-Port B Loopback Plug Wiring Diagram

25-pin Port A-to-A Port B-to-B Loopback Plug

If your system has a single communication port to connect it to peripherals, use these wiring instructions for making a male 25-pin loopback plug for that communication port.
96-Pin Female Loopback Connector

This 96-pin connector can be ordered from Sun (part number 370-1366).
FIGURE A-7  96-Pin Female Loopback Connector Wiring Diagram

Materials:
- PCR-E96FA(1)
- PCS-E96LKP A(1)
- 3751 Metal Plug(1)
- (9563K42)
- AWG28 Madison Cable(8" long)
  UL/CSA Approved

Connect:

<table>
<thead>
<tr>
<th>First connector</th>
<th>Second connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pins 4 and 12</td>
<td>Pin 77</td>
</tr>
<tr>
<td>Pin 5</td>
<td>Pin 20</td>
</tr>
<tr>
<td>Pins 6</td>
<td>Pin 36</td>
</tr>
<tr>
<td>Pin 10</td>
<td>Pin 11</td>
</tr>
<tr>
<td>Pins 13</td>
<td>Pin 16</td>
</tr>
<tr>
<td>Pin 14</td>
<td>Pin 15</td>
</tr>
<tr>
<td>Pin 18</td>
<td>Pin 19</td>
</tr>
<tr>
<td>Pin 21</td>
<td>Pin 24</td>
</tr>
<tr>
<td>Pin 28</td>
<td>Pin 60</td>
</tr>
<tr>
<td>Pin 29</td>
<td>Pin 68</td>
</tr>
<tr>
<td>Pin 30</td>
<td>Pin 34</td>
</tr>
<tr>
<td>Pin 37</td>
<td>Pin 40</td>
</tr>
<tr>
<td>Pin 38</td>
<td>Pin 39</td>
</tr>
<tr>
<td>Pin 42</td>
<td>Pin 43</td>
</tr>
<tr>
<td>Pin 45</td>
<td>Pin 48</td>
</tr>
<tr>
<td>Pin 46</td>
<td>Pin 47</td>
</tr>
<tr>
<td>Pin 52</td>
<td>Pin 78</td>
</tr>
<tr>
<td>Pin 53</td>
<td>Pin 55</td>
</tr>
<tr>
<td>Pin 54</td>
<td>Pin 75</td>
</tr>
<tr>
<td>Pin 58</td>
<td>Pin 59</td>
</tr>
<tr>
<td>Pin 61</td>
<td>Pin 64</td>
</tr>
<tr>
<td>Pin 62</td>
<td>Pin 63</td>
</tr>
<tr>
<td>Pin 66</td>
<td>Pin 67</td>
</tr>
<tr>
<td>Pin 69</td>
<td>Pin 72</td>
</tr>
<tr>
<td>Pin 76</td>
<td>Pin 79</td>
</tr>
<tr>
<td>Pin 82</td>
<td>Pin 83</td>
</tr>
<tr>
<td>Pin 85</td>
<td>Pin 88</td>
</tr>
<tr>
<td>Pin 86</td>
<td>Pin 87</td>
</tr>
<tr>
<td>Pin 90</td>
<td>Pin 91</td>
</tr>
<tr>
<td>Pin 93</td>
<td>Pin 96</td>
</tr>
<tr>
<td>Pin 94</td>
<td>Pin 95</td>
</tr>
</tbody>
</table>
## 96-Pin Female Special Loopback Connector

This 96-pin connector can be ordered from Sun (part number 370-1381).

![96-Pin Female Special Loopback Connector Wiring Diagram](image)

<table>
<thead>
<tr>
<th>Connect:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>First Connector</td>
<td>Second Connector</td>
</tr>
<tr>
<td>pin 3</td>
<td>pin 4</td>
</tr>
<tr>
<td>pin 5</td>
<td>pin 7</td>
</tr>
<tr>
<td>pins 8 and 9</td>
<td>pin 12</td>
</tr>
<tr>
<td>pin 10</td>
<td>pin 11</td>
</tr>
<tr>
<td>pin 13</td>
<td>pin 14</td>
</tr>
<tr>
<td>pin 15</td>
<td>pin 17</td>
</tr>
<tr>
<td>pins 18 and 19</td>
<td>pin 22</td>
</tr>
<tr>
<td>pin 20</td>
<td>pin 21</td>
</tr>
<tr>
<td>pin 27</td>
<td>pin 28</td>
</tr>
<tr>
<td>pin 29</td>
<td>pin 31</td>
</tr>
<tr>
<td>pins 32 and 33</td>
<td>pin 36</td>
</tr>
<tr>
<td>pins 34</td>
<td>pin 35</td>
</tr>
<tr>
<td>pins 37</td>
<td>pin 38</td>
</tr>
<tr>
<td>pin 39</td>
<td>pin 41</td>
</tr>
<tr>
<td>pins 42 and 43</td>
<td>pin 46</td>
</tr>
<tr>
<td>pin 44</td>
<td>pin 45</td>
</tr>
<tr>
<td>pin 51</td>
<td>pin 52</td>
</tr>
<tr>
<td>pin 53</td>
<td>pin 55</td>
</tr>
<tr>
<td>pins 56 and 57</td>
<td>pin 60</td>
</tr>
<tr>
<td>pin 58</td>
<td>pin 59</td>
</tr>
<tr>
<td>pin 61</td>
<td>pin 62</td>
</tr>
<tr>
<td>pin 63</td>
<td>pin 65</td>
</tr>
<tr>
<td>pins 66 and 67</td>
<td>pin 70</td>
</tr>
<tr>
<td>pin 68</td>
<td>pin 69</td>
</tr>
<tr>
<td>pin 75</td>
<td>pin 76</td>
</tr>
<tr>
<td>pin 77</td>
<td>pin 79</td>
</tr>
<tr>
<td>pins 80 and 81</td>
<td>pin 84</td>
</tr>
<tr>
<td>pin 82</td>
<td>pin 83</td>
</tr>
<tr>
<td>pin 85</td>
<td>pin 86</td>
</tr>
<tr>
<td>pin 87</td>
<td>pin 89</td>
</tr>
<tr>
<td>pins 90 and 91</td>
<td>pin 94</td>
</tr>
<tr>
<td>pin 92</td>
<td>pin 93</td>
</tr>
</tbody>
</table>

Materials:
- PCR-E96FA(1)
- PCS-E96LKP A(1)
- 9563K999 Cap(1)
- Madison Cable(6" long)
- (#28 SCSI UL/CSA Approved)

Open Contacts:
1, 2, 6, 16, 23, 24, 25, 26, 30, 40, 47, 48, 49, 50, 54, 64, 71, 72, 73, 74, 78, 88, 95, 96
37-Pin RS-449 Loopback Cable

Use these wiring instructions for a loopback cable for two 37-pin RS-449 synchronous ports.

FIGURE A-9  37-Pin RS-449 Loopback Cable Wiring Diagram
37-Pin RS-449 Loopback Plug

Use these wiring instructions for making a male 37-pin RS-449 loopback plug. This connector is also available from Sun (part number 530-1430).

FIGURE A-10 37-Pin RS-449 Loopback Plug Wiring Diagram
9-pin Male Single-Port Loopback Plug

Use these wiring instructions for male 9-pin RS-232 and RS-423 single-port loopback plugs.

![Male Loopback Plug Wiring Diagram](image)

Connect:
First connector to Second connector

- Pin 2 to Pin 3
- Pin 1 to Pins 4 and 6
- Pin 7 to Pin 8

FIGURE A-11 9-Pin Male Single-Port Loopback Plug Wiring Diagram

9-pin Female Single-Port Loopback Plug

Use these wiring directions for female 9-pin RS-232 and RS-423 single-port loopback plugs. Use this loopback plug with the pcmciatest.

![Female Loopback Plug Wiring Diagram](image)

Connect:
First connector to Second connector

- Pin 2 to Pin 3
- Pin 1 to Pins 4 and 6
- Pin 7 to Pins 8 and 9

FIGURE A-12 9-Pin Female Single-Port Loopback Plug Wiring Diagram
9-Pin to 25-Pin Port-to-Port Loopback Cable

Use these wiring instructions for a 9-pin RS-232 and RS-423 port to 25-pin RS-232 and RS-423 port loopback cables. Both connectors are male.

<table>
<thead>
<tr>
<th>Connect:</th>
<th>First connector to Second connector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pin 2</td>
<td>Pin 2</td>
</tr>
<tr>
<td>Pin 3</td>
<td>Pin 3</td>
</tr>
<tr>
<td>Pin 8</td>
<td>Pin 4</td>
</tr>
<tr>
<td>Pin 7</td>
<td>Pin 5</td>
</tr>
<tr>
<td>Pin 5</td>
<td>Pin 7</td>
</tr>
<tr>
<td>Pin 4</td>
<td>Pins 6 and 8</td>
</tr>
<tr>
<td>Pins 1 and 6</td>
<td>Pin 20</td>
</tr>
</tbody>
</table>

**FIGURE A-13** 9-Pin to 25-Pin Port-to-Port Loopback Cable Wiring Diagram
9-Pin to 9-Pin Port-to-Port Loopback Cable

Use these wiring instructions for 9-pin RS-232 and RS 423 port to 9-pin RS-232 and RS-423 port loopback cables. Both connectors are male.

**FIGURE A-14** 9-Pin to 9-Pin Port-to-Port Loopback Cable Wiring Diagram

Please note that this cable has no Sun part number assigned to it.

NT to TE Loopback Cable

Using two standard RJ-45 connectors, and connect pin 1 to pin 1, pin 2 to pin 2, and so on, for all pins. This loopback is a “straight-through” connection.
Twisted-Pair Ethernet (TPE) Loopback Cable For Fast Ethernet

Use the following wiring instructions for standard RJ-45 connectors for Fast Ethernet. This loopback cable is used in netlbtest for eri devices.

![Diagram of wiring instructions](image)

TPE Loopback Cable For Gigabit Ethernet

Use the following wiring instructions for RJ-45 connectors for Gigabit Ethernet. This loopback cable is used in netlbtest for ge devices.
FIGURE A-16 TPE Loopback Cable Wiring Diagram for Gigabit Ethernet

Connect:
- Pin 1 to Pin 3
- Pin 2 to Pin 6
- Pin 4 to Pin 7
- Pin 5 to Pin 8

Diagram showing the wiring connections as follows:
- Pin 1 connected to Pin 3
- Pin 2 connected to Pin 6
- Pin 4 connected to Pin 7
- Pin 5 connected to Pin 8
Index

NUMERICS
25-pin port A-to-A port B-to-B loopback plug, 370
25-pin port A-to-port B loopback plug, 370
25-pin RS-232 loopback plug, 367
25-pin RS-232 port-to-port loopback cable, 367
64-bit and 32-bit tests, 3
8-pin loopback plug, 369
8-pin to 8-pin loopback cable, 368

atmtest, SunATM adapter test, 31
command line syntax, 35
options, 32
test modes, 35

Audio Test, audio subsystem test, 37
audiotest subsystem test, audiotest, 37
audiotest, audiotest subsystem test
command line syntax, 41
options, 39

A
Acceleration tests of tcxtest, 341
accessing SunVTS, 3
Advanced Frame Buffer test, afbtest, 11
afbtest, Advanced Frame Buffer test, 11
command line syntax, 20
options, 12
test modes, 19
AFX Protocol tests of tcxtest, 341
Alarm Card, Alarm Card test, 23
command line syntax, 27
options, 23
test modes, 27
Alarm Card, Alarm Card test
options, 25
Alm test, See sptest, 305
Apply to All button, 5
Apply to Group button, 5
arguments
standard, 6

B
Bidirectional parallel port test, bpptest, 45
bpptest, Bidirectional parallel port test, 45
command line syntax, 48
options, 46
requirements, 45
test modes, 47
buttons
Apply to All, 5
Apply to Group, 5
Cancel, 6
Reset, 6

C
cache consistency test, mpconstest, 215
Cancel button, 6
CDE, (Common Desktop Environment), xvii
CD-ROM test, cdtest, 49
cdtest, Compact Disc test, 49
and Volume Management, 49
command line syntax, 52
options, 50
test modes, 52
cg14 frame buffer card test, cg14test, 55
cg14test, CG14 test, 55
command line syntax, 66
options, 65
test groups, 55
test modes, 66
cg6 test, cg6, 67
cg6, CG6 frame buffer test, 67
and swap requirements, 71
command line syntax, 71
options, 70
subtests, 68
test modes, 70
circuit test, i2ctest, 179
Colormap and Cursor tests of tcxtest, 341
Command line execution, 6
command line options, 6
command line syntax, 289
Common Desktop Environment (CDE), xvii
Compact Disc (CD) test, cctest, 49
cpu Power Management test, cpupmtest, 79
CPU stress test, systest, 331
CPU tests
cputest, 73
fputest, 155
mptest, 223
multiprocessor test mptest, 223
systest, 331
cpupmtest, CPU Power Management test, 79
command line syntax, 81
test modes, 81
cpupmtest, CPU Power Management test
options, 79
cputest, CPU test, 73, 163
command line syntax, 77
options, 74
test modes, 77
cputest, cpu test
options, 74

D
disk drive test, 83
disktest, disk and floppy test, 83
command line syntax, 89
options, 86
subtests, 84
test modes, 89
dpmtest, Sun Fire 880 FC-AL Disk Backplane
test modes, 93
dpmtest, Sun Fire 880 FC-AL Disk Backplane
test, 91
DVD drive test, dvdtest, 95
dvdtest, DVD drive test, 95
command line syntax, 97
options, 95
test modes, 97

E
ECC memory errors, locating with pmemtest, 251
ecpp(7) IEEE 1284 parallel printer port device
test, 99
ecpptest, Parallel port printer test, 99
command line syntax, 103
options, 101
requirements, 99
subtests, 100
enatest, Sun Enterprise Array test, 105
command line syntax, 113
fault isolation, 112
options, 110
status, 109
test modes, 113
enctest, StorEdge 1000 test, 115
command line syntax, 118
options, 116
test modes, 118
env2test, I2C bus test, 123
options, 123
test modes, 125
env3test, Environmental Hardware test, 127
command line syntax, 129
env4test, I2C bus test, 131
    options, 132
test modes, 135
env5test, Environmental test, 137
env5test, environmental test
    command line syntax, 140
    options, 138
test modes, 140
test requirements, 137
Environmental hardware tests
env3test, 127
evttest, 119
environmental test, env5test, 137
evttest, I2C bus test, 119
    command line syntax, 122
    options, 120
Ethernet hardware test, nettest, 229
ethernet loopback test, netlbtest, 235
exercising SSA fiber channel with pntest, 245
Expert3D Frame Buffer test, ifbstest, 183

F
Fast Frame Buffer test, f bstest, 145
fbtest frame buffer test, 141
    command line syntax, 143
    options, 142
test modes, 143
f bstest, Fast frame buffer test, 145
    cautions and notes, 145
    command line syntax, 153
    options, 146
test modes, 152
fiber optic test, nettest, 229
floating point unit test, fputest, 155
floppy drive and disk test, disktest, 83
fputest, floating Point Unit test, 155
    command line syntax, 158
    options, 156
frame buffer
    multiple, 9
testing, 9
Frame Buffer Memory tests tctest, 341
frame buffer test, generic, fbstest, 141
Frame buffer tests
cgl4test, 55
cg6, 67
frame buffer tests
gfxtest, 175
fwcamtest, Fire Wire Camera test, 159
gfxtest
    command line syntax, 162
    options, 160
test modes, 162
G
gfbtest Options, 164, 166
gfbtest Test Parameters Options Dialog Box, 165
gfbtest Test Requirements, 164
gfxtest
    cautions and notes, 175
gfxtest
    command line syntax, 178
    options, 176
test modes, 178
gfxtest, PGX32 Frame butter test, 175
graphical user interface (GUI), xvii
GUI (graphical user interface), xvii
GX & GX+ test, cg6, 67

H
hard drive and floppy test, disktest, 83
HDLC and SDLC test (for the HSI), sunlink, 315
HSI board test, sunlink, 315

I
I2C bus test, env2test, 123
I2C bus test, env4test, 131
I2C bus test, envtest, 119
I2C Bus tests
    i2ctest, 179
i2ctest, I2C bus test, 179
    command line syntax, 181
    options, 180
test modes, 181
test requirements, 179
ifbstest, Expert3D Frame Buffer test, 183
    command line syntax, 189
options, 185
  test modes, 189
Test Requirements, 183
ifptest, PCI FC_AL card test
  command line syntax, 195
  options, 192
  test modes, 194
ifptest, PCI FC_AL card test, 191
Installation directory, 2
Internet control message protocol (ICMP) and
  nettest, 229
interprocess communication protocols, 2
  IPC (interprocess communication), 2
  IPC tests
    disktest, 83

K
  keyboard test, usbkbtest, 349

L
  ll1dcachetest, level 1 data cache test, 197
    command line syntax, 199
    options, 197
    test modes, 199
  ll2dcachetest, level 2 cache test, 201
    command line syntax, 203
    options, 201
    test modes, 203
  level 1 data cache test, ll1dcachetest, 197
  level 2 cache test, ll2dcachetest, 201
  LOMLite alarm test, lomlitetest, 205
  lomlitetest, LOMLite alarm test, 205
    command line syntax, 208
    options, 207
    requirements, 205
    subtests, 206
    test modes, 208
loopback connectors
  25-pin port A-to-A port B-to-B plug, 370
  25-pin RS-232 plug, 367
  25-pin RS-232 port-to-port cable, 367
  8-pin plug, 369
  8-pin to 8-pin cable, 368
  port A-to-port B plug, 370
See Appendix A

M
  M64 video board test, m64test, 209
  m64test, M64 Video test, 209
    cautions and notes, 209
    command line syntax, 213
    options, 210
    subtests, 209
    test modes, 212
  Major test categories, 1
  Media tests
    cdtest, 49
    disktest, 83
    tapetest, 335
  Memory errors detected by vmemtest, 353
  Memory tests
    pmemtest, 251
    vmemtest, 353
  mouse
    movement, can cause test failure, 9
  mpconstest, cache consistency test, 215
  mpconstest, cache consistency test
    command line syntax, 221
    options, 218
    subtests, 217
    test modes, 221
  mptest, multiprocessor test, 223
    command line syntax, 227
    options, 224
    test modes, 227
  multiprocessor test, mptest, 223

N
  netlbtest, ethernet loopback test, 235
    command line syntax, 238
    options, 236
    test modes, 238
    test requirements, 235
  nettest, network hardware test, 229
    command line syntax, 233
    options, 230
test modes, 232
Network tests
netlbtest, 235
nettest, 229

P
parallel port test, bpptest, 45
parallel printer port test ecpptest, 99
parallel/serial port test, spiftest, 299
parity errors, locating with pmemtest, 251
PCI FC_AL card test, ifptest, 191
PCI tests
atmtest, 31
pcsertest PCMCIA modem card test, 241
command line syntax, 243
loopback requirement, 241
options, 242
test mode, 243
Peripheral tests
cdtest, 49
disktest, 83
tapetest, 335
usbkbtest, 349
physical memory test, pmemtest, 251
pcsertest PCMCIA modem card test, 241
command line syntax, 243
loopback requirement, 241
options, 242
test mode, 243
Printer tests
bpptest, 45
ecpptest, 99
protocols, interprocess communication, 2

Q
qlctest, Qlogic 2202 PC / AL Crystal test, 257
command line syntax, 262
test modes, 262
quad ethernet test, nettest, 229

R
register testing with fputest, 155
remote
testing, 10
Remote System Control (RSC) test rsctest, 265
Remote System Control test
rsctest, 265
Requirements, 2
Reset button, 6
rsctest
Remote System Control test modes, 270
rsctest, Remote System Control test, 265
Command line syntax, 270
rsctest: Remote System Control test
options, 268

S
saiptest, Serial Asynchronous Interface (PCI) test, 273
available ports, 276
command line syntax, 277
hardware requirements, 273
saiptest, Serial Asynchronous Interface (PCI) test
options, 274
test modes, 277
SBus DMA circuitry, test with bpptest, 45
SBus printer card test, bpptest, 45
SBus tests
atmtest, 31
bpptest, 45
scitest, Cluster networking hardware test, 281
command line syntax, 283
options, 282
test modes, 283
scitest, SunFireLink Interconnect test, Cluster networking hardware test, 359
SCSI Environmental Sensing card test, sentest, 291
SCSI Environmental Sensing card test, sentest, 291
SCSI tests
cdtest, 49
disktest, 83
dvdttest, 95
sentest, SCSI Environmental Sensing card test, 291
command line syntax, 294
options, 292
test modes, 293
verification, 291
Serial Asynchronous Interface test, saiptest, 273
serial port test, sptest, 305
serial/parallel port test, spiftest, 299
Smart Card test, Caution sctest, 285
Smart Card test, sctest, 285, 289
command line syntax, 289
options, 287
requirements, 285
subtests, 287
test modes, 289
test parameters options, 288
test parameters options Dialog Box, 288
SOC+ host adapter card test, socaltest, 295
socaltest, Soc+ card test, 295
command line syntax, 298
options, 296
test modes, 298
SPARCstation 5 S24 frame buffer test, tcxtest, 341
SPARCstation tests
audiotest, 37
SPARCstation VSIMM test, cg14test, 55
SPARCstation4 TCX frame buffer test, tcxtest, 341
SPARCstorage Array controller test, plntest, 245
spiftest, serial/parallel port test, 299
available ports, 301
command line syntax, 303
hardware requirements, 299
options, 300
test modes, 303
sptest, serial port test, 305
command line syntax, 313
options, 307
requirements, 307
test modes, 312
tests, 306
SSA
exercising fiber channel with plntest, 245
probing for controller devices, 246
SSA controller test, plntest, 245
SSA fault isolation with plntest, 245
standard
command line arguments, 6
usage, 6
standard arguments, 6
Storage subsystem tests
enatest, 105
StorEdge 1000 enclosure test, enctest, 115
Sun Enterprise 450 bus test, envtest, 119
Sun Enterprise Cluster 2.0 Network Hardware test
scitest, 281
Sun Enterprise Cluster networking hardware test, scitest, 281
Sun Enterprise Network Array test, 105
Sun Enterprise Network Array test, enatest, 105
Sun Fire 880 FC-AL Disk Backplane test, dpmtest, 91
SunATM adapter test, atmtest, 31
SunFireLink Interconnect tests
wrsmttest, 359
sunlink, HSI/S board test, 315
command line syntax, 318
loopback connectors, 317
options, 316
requirements, 315
test modes, 318
sunpci2test, Sun PCi2 test, 321
command line syntax, 323
options, 322
test requirements, 321
SunVTS
accessing, 3
interfaces
CDE, 3
OPEN LOOK, 3
TTY, 3
sutest, Super I/O test, 325
command line syntax, 328
options, 326, 327
test modes, 328
systest, CPU stress test, 331
command line syntax, 333
options, 331
test modes, 333
T

tapetest, tape drive test, 335
    command line syntax, 339
    options, 335
    test modes, 339
    test requirements, 335
tcxtest, Fast SBus Video card test, 341
tcxtest, S24 Frame Buffer SBus card test
test groups, 341
tcxtest, S24 Frame Buffer SBus card test, 341
    command line syntax, 346
    options, 344
    subtests, 343
    test modes, 346
test options, 6
Test Parameter menu, 4
testing
    remotely, 10
Tests
    32-bit and 64-bit tests, 3
Tests, overall description of, 2
Test-specific arguments, 8
Test-specific menu, 4
Test-specific options, 5
TGX test, cg6, 67
token ring test, nettest, 229

U

usbkbtset, USB keyboard test, 349
    command line syntax, 351
    options, 349
    test modes, 351

V

Video test
cg6, 67
Video tests
cgl4test, 55
fbtest, 141
ffbtset, 145
gfxtset, 175
m64tset, 209
virtual memory test, vmemtset, 353

vmemtset, virtual memory test, 353
    command line syntax, 357
    options, 354
    swap space, and, 353
    test modes, 357
Volume Management and disktest, 84
VSIMM test, cg14test, 55

W

window
    locking disabled, 10
    locking enabled, 10
wrsmtset, SunFireLink Interconnect test, Cluster networking hardware test
    command line syntax, 363
    options, 362
    test modes, 363

Z

zs test, sprtest, 305