



# SunVTS™ 5.1 User's Guide

---

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

Part No. 816-5144-10  
September 2002, Revision B

Send comments about this document to: [docfeedback@sun.com](mailto:docfeedback@sun.com)

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

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

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

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

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

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

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

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

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in the Sun Microsystems, Inc. license agreements and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct. 1998), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

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

---

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

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

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

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

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

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

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

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

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



Adobe PostScript

# Contents

---

**Preface vii**

**1. SunVTS Overview 1-1**

- 1.1 What Is SunVTS? 1-1
- 1.2 SunVTS Features 1-2
- 1.3 SunVTS Architecture 1-6

**2. Installing and Removing SunVTS 2-1**

- 2.1 SunVTS Packages 2-2
- 2.2 Installation Requirements 2-3
- 2.3 Installing SunVTS 2-3
  - ▼ To Install SunVTS With the `pkgadd` Command 2-3
  - ▼ To Set Up Access to the SunVTS Man Pages 2-5
  - ▼ To Perform Additional Setup When SunVTS Is Installed in a Directory Other Than `/opt` 2-6
- 2.4 SunVTS Security 2-7
- 2.5 SunVTS Environmental Variables 2-12
- 2.6 Additional Instructions for Localized Environments 2-12
  - ▼ To Run SunVTS in English, in a Non-English Environment 2-13
  - ▼ To Set Up the GUI `resource` File for Localized Environments 2-13
- 2.7 Adding a Custom Test 2-14

- ▼ To Add a Custom Test 2-14
- 2.8 Removing SunVTS 2-16
  - ▼ To Remove SunVTS With the `pkgrm` Command 2-16
- 3. Starting SunVTS 3-1**
  - 3.1 SunVTS Runtime Considerations 3-1
  - 3.2 Requirements for Starting SunVTS 3-2
  - 3.3 Overview of SunVTS Procedures 3-3
  - 3.4 Preparing Devices for Testing 3-4
  - 3.5 Starting SunVTS on a Local System 3-5
  - 3.6 Starting SunVTS on a Remote System 3-9
  - 3.7 Quitting SunVTS 3-13
    - ▼ To Quit SunVTS 3-13
  - 3.8 Displaying Devices With the `vtsp` Utility 3-13
    - ▼ To Display the Devices for a Local System 3-13
    - ▼ To Display Devices of a Remote System 3-15
- 4. Using the SunVTS CDE User Interface 4-1**
  - 4.1 Getting Started With the SunVTS CDE UI 4-1
  - 4.2 Using Additional Features With the SunVTS CDE UI 4-18
  - 4.3 Scheduling Test Sequences With the Deterministic Scheduler 4-30
- 5. Using the SunVTS TTY User Interface 5-1**
  - 5.1 Getting Started With the SunVTS TTY UI 5-1
  - 5.2 Navigating the SunVTS TTY UI 5-4
    - ▼ To Configure SunVTS for a Test Session 5-4
    - ▼ To Run, Monitor, and Stop a Test Session 5-13
    - ▼ To Review and Reset Test Session Results 5-14
  - 5.3 Using Additional Features With the SunVTS TTY UI 5-17

5.4 Scheduling Test Sequences With the Deterministic Scheduler 5-27

**A. SunVTS Window and Dialog Box Reference A-1**

**B. Frequently Asked Questions B-1**

**C. Creating and Editing Generic Option Files for Advanced Users C-1**

**D. SunVTS Global Configuration File `sunvts.conf` D-1**



# Preface

---

SunVTS™ is Sun's Validation test suite. SunVTS is a comprehensive diagnostic tool that tests and validates Sun hardware by verifying the connectivity and proper functioning of most hardware controllers, devices, on Sun platforms.

The primary audience of this manual is hardware testing and verification personnel, qualified service-trained maintenance providers, and advanced system end users.

This book describes how to install, configure, and use SunVTS.

---

## How This Book Is Organized

Chapter 1 provides an overview of SunVTS and the SunVTS user interfaces, architecture, and test modes.

Chapter 2 describes the SunVTS packages and installation procedures.

Chapter 3 covers the different methods used to start SunVTS.

Chapter 4 describes how to configure, start, monitor, and review a test session with the SunVTS CDE UI.

Chapter 5 describes how to configure, start, monitor, and review a test session with the SunVTS TTY UI.

Appendix A provides window and dialog box look-up tables.

Appendix B provides answers to SunVTS frequently asked questions.

Appendix C provides detailed information about editing the generic option file for advanced users.

Appendix D provides detailed information for configuring the SunVTS global configuration file, `sunvts.conf`.

---

## Using UNIX Commands

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

See one or more of the following for this information:

- *Solaris Handbook for Sun Peripherals*
- Online documentation for the Solaris™ software environment available at: <http://docs.sun.com>
- Other software documentation that you received with your system

---

## Typographic Conventions

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



---

## Shell Prompts

Shell	Prompt
C shell	<i>machine-name%</i>
C shell superuser	<i>machine-name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

---

## Related Documentation

Application	Title	Part Number
Reference	<i>SunVTS 5.1 Test Reference Manual</i>	816-5145-10
Quick Reference	<i>SunVTS Quick Reference Card</i>	816-5146-10

---

## Accessing Sun Documentation

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

<http://www.sun.com/documentation>

---

# Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can email your comments to Sun at:

`docfeedback@sun.com`

Please include the part number (816-5144-10) of your document in the subject line of your email.

# SunVTS Overview

---

This chapter covers:

- “What Is SunVTS?” on page 1-1
- “SunVTS Features” on page 1-2
- “SunVTS Architecture” on page 1-6

---

## 1.1 What Is SunVTS?

SunVTS is Sun Validation test suite. SunVTS is a comprehensive diagnostic tool that tests and validates Sun hardware by verifying the connectivity and functionality of most hardware controllers and devices on Sun platforms.

---

**Note** – SunVTS is only supported on the Sun SPARC™ platforms running the Solaris operating environment.

---

SunVTS can be tailored to run on various types of machines ranging from desktops to servers with modifiable test instances and processor affinity features.

SunVTS supports testing in both 32-bit and 64-bit Solaris operating environments; automatically determines the operating environment (32-bit or 64-bit); and initiates the appropriate tests.

Use SunVTS to validate a system during development, production, receiving inspection, troubleshooting, periodic maintenance, and system or subsystem stressing.

SunVTS has a sophisticated graphical user interface (UI) that provides test configuration and status monitoring. The user interface can be run on one system to display the SunVTS testing of another system on the network. SunVTS also provides a TTY-mode interface for situations in which running a graphical UI is not possible.

---

## 1.2 SunVTS Features

### 1.2.1 New Features for SunVTS 5.1

The SunVTS 5.1 software is compatible with Solaris 8 2/02, Solaris 9, and Solaris 9 9/02 operating environments. Two new test modes (Online and Exclusive) were added for this release of SunVTS; refer to “SunVTS Test Modes” on page 1-5 for descriptions of each. Also, creating and editing a generic configuration option file is now supported and is described in Appendix C. Additionally, online help is now available by selecting Help/Online Help from the main CDE Diagnostic window or the TTY UI.

### 1.2.2 Test Categories

SunVTS comprises many individual tests for testing a wide range of Sun products and peripherals.

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

- Audio tests
- Communication (Serial and Parallel) tests
- Graphic/Video tests
- Memory tests
- Network tests
- Peripherals (Disks, Tape, CD-ROM, Printer, Floppy) tests
- Processor tests
- Storage tests

You need to select the proper test modes and options to maximize the effectiveness of SunVTS. This book covers the overall test configuration modes and options of the application. For information regarding the configuration of individual tests, refer to the *SunVTS Test Reference Manual*.

## 1.2.3 SunVTS User Interfaces

You can run SunVTS from the following user interfaces:

- Graphical user interface—use the CDE environment. FIGURE 1-1 shows the SunVTS CDE interface.
- TTY interface—You can run SunVTS from a terminal, a shelltool, or a modem attached to a serial port using the TTY interface.
- Individual test execution—run individual tests from the command line.

---

**Note** – As of SunVTS 5.0, the SunVTS OPEN LOOK user interface is no longer supported.

---

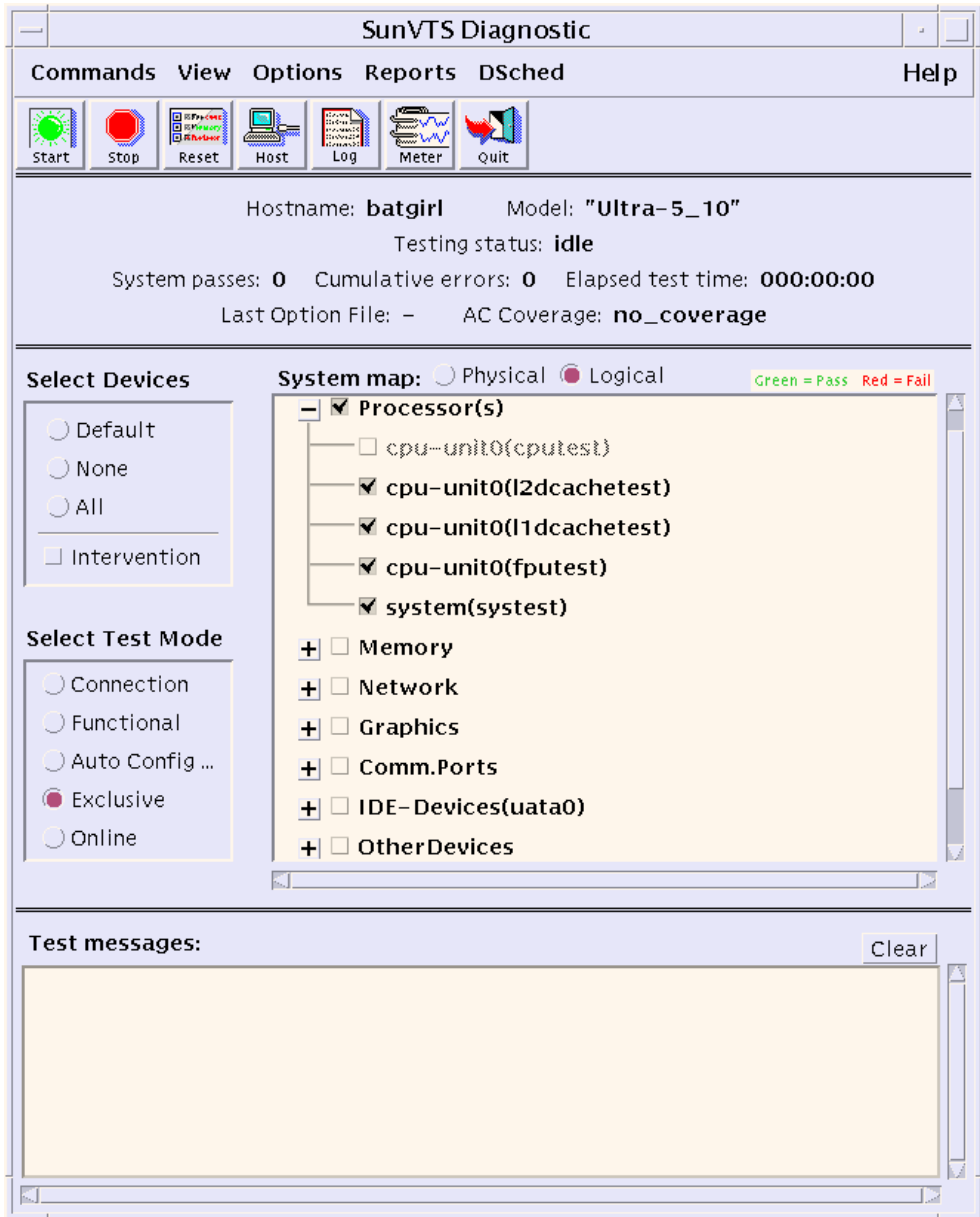


FIGURE 1-1 SunVTS CDE Main Window

## 1.2.4 SunVTS Test Modes

Based on how you configure SunVTS, the test session runs in one of the following test modes:

- **Connection test mode**—provides a low-stress, quick testing of the availability and connectivity of selected devices. These tests are nonintrusive, meaning they release the devices after a quick test, and they do not place a heavy load on system activity.
- **Functional test mode**—provides robust testing of your system and devices. It uses your system resources for thorough testing and it assumes that no other applications are running.
- **Exclusive test mode**—enables performing the tests that require no other `sunvts` tests or applications running at the same time.
- **Online test mode**—enables users to perform `sunvts` testing while customer applications are running.
- **Auto Config test mode**—simplifies the SunVTS configuration process by automatically assigning a predetermined set of test options.

## 1.2.5 Security

SunVTS has two security mechanisms that you choose to implement when you install SunVTS:

- **Basic Security**—a local file is maintained that lists the valid users, groups, and hosts permitted to use SunVTS. This level of security does not provide secure network authentication and should not be used in an environment where network security is an issue.
- **SEAM Security**—uses the Kerberos-based, Sun Enterprise Authentication Mechanism™ (SEAM) to provide secure user authentication, data integrity and privacy, for transactions over networks. To use the SunVTS Kerberos security, you must have the SEAM software installed in your networked environment.

The SunVTS installation process prompts you to specify which security mechanism you want to use. You must use one or the other, and the SEAM security implementation is the default if you press the Return key through the installation questions.

## 1.3 SunVTS Architecture

The SunVTS architecture is divided into three components: the SunVTS kernel, the user interface, and the collection of hardware tests. FIGURE 1-2 is a block diagram representing the SunVTS architecture.

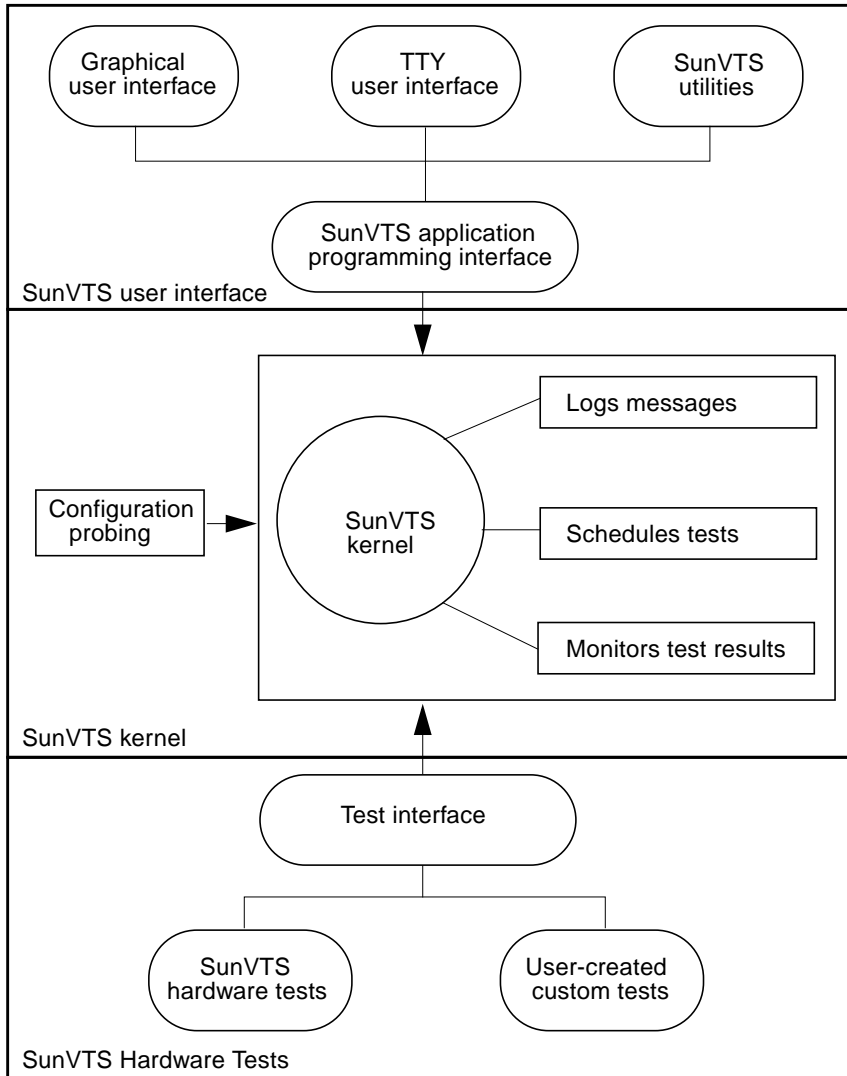


FIGURE 1-2 SunVTS Architecture



## 1.3.1 SunVTS Kernel

The following activities are scheduled and monitored by the kernel:

- Probing and saving the test system's hardware configuration at start-up
- Logging messages reported from tests
- Maintaining the status of all running tests
- Monitoring the status requests and controlling the commands from a user interface or other applications
- Scheduling the tests that are enabled by the user
- Running as a background process (daemon)

There are two versions of the SunVTS kernel:

- 32-bit kernel— `/opt/SUNWvts/bin/vtsk`
- 64-bit kernel— `/opt/SUNWvts/bin/sparcv9/vtsk`

## 1.3.2 SunVTS User Interface

The SunVTS user interface is separate from the SunVTS kernel. The user interface communicates with the SunVTS kernel through an application programming interface (API). This gives SunVTS the ability to run the appropriate interface (CDE, or TTY) based on the environment of the system. It also allows the user interface to run on a system other than the system under test.

## 1.3.3 SunVTS Hardware Tests

Many separate tests make up the collection of tests in the SunVTS application. Each test is a separate process from the SunVTS kernel.

When SunVTS is started, the SunVTS kernel automatically probes the system kernel to determine the installed hardware devices and displays the testable devices in the SunVTS UI. This provides a quick check of your hardware configuration, and only those tests applicable to your system are displayed.

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.

The SunVTS kernel and most tests support 32-bit and 64-bit Solaris operating environments.



## Installing and Removing SunVTS

---

This chapter describes how to install and remove the SunVTS diagnostic program.

- “SunVTS Packages” on page 2-2
- “Installing SunVTS” on page 2-3
- “To Set Up Access to the SunVTS Man Pages” on page 2-5
- “To Perform Additional Setup When SunVTS Is Installed in a Directory Other Than /opt” on page 2-6
- “SunVTS Security” on page 2-7
- “Additional Instructions for Localized Environments” on page 2-12
- “Removing SunVTS” on page 2-16
- “Adding a Custom Test” on page 2-14

---

## 2.1 SunVTS Packages

The SunVTS software is installed from the packages shown in the following table.

TABLE 2-1 SunVTS Packages

Package Name	Description
SUNWvts	Contains the SunVTS kernel, user interface, and the collection of 32-bit binary tests. This is the main diagnostic package. You must install it to run SunVTS.
SUNWvtsx	Provides SunVTS 64-bit binary tests and kernel. This package should be loaded on systems that support 64-bit execution, but it is not necessary for SunVTS 32-bit functionality. This package cannot be installed on systems that do not have the 64-bit Solaris operating system installed.
SUNWvtsmn	Contains the SunVTS man pages. These files are installed in the <code>/opt/SUNWvts/man</code> directory by default, and you may need to update your <code>MANPATH</code> variable (described in this chapter). Installing these man pages is optional; however, they provide helpful information about SunVTS commands.

**Note 1:** The `configd` packages (`SUNweswsa`, `SUNwsycfd`, `SUNwesnta`, and `SUNweswga`) are no longer needed and are no longer supplied for physical mapping support.

**Note 2:** As of SunVTS 4.3, the SunVTS online testing capability that was initiated using the `vtsui.online` command is no longer available. The `SUNWodu` package that provides this online testing function is no longer provided.

Online Diagnostic testing of Sun systems is now available through the Sun Management Center software using the Sun Hardware Diagnostic Suite add-on software. See <http://www.sun.com/sunmanagementcenter> for details.

**Note 3:** As of SunVTS 5.0, the SunVTS OPEN LOOK user interface (UI) is no longer supported. The SunVTS OPEN LOOK tests (`sundials` and `sunbuttons`) are discontinued. For full feature support, use the SunVTS CDE or TTY UI. Refer to the Solaris “End-of-Software Support Statements” section of the Solaris operating environment release notes for the latest end of support news.

---

SunVTS packages are located on the Sun Computer Systems Supplement CD that ships with the Solaris release.

---

## 2.2 Installation Requirements

- The Solaris 9 operating environment must be installed (with the Entire Distribution software group [SUNWCa11] at minimum), and booted to multi-user level (run level 3).
- You must be superuser.
- There must be at least 62 MB of available disk space in the partition where SunVTS will reside. The `/opt` directory is the default installation location.

---

**Note** – Refer to Appendix B for information about which revisions of SunVTS are supported on different Solaris operating environment releases.

---

---

## 2.3 Installing SunVTS

There are several utilities available for installing packages. This chapter describes how to use the `pkgadd` utility to load SunVTS from a local CD-ROM drive. For information on how to install packages using other installation methods, see the *Solaris 9 Sun Hardware Platform Guide*.

### ▼ To Install SunVTS With the `pkgadd` Command

1. Log in to the system and become superuser:

```
% su
```

2. Check to see if any SunVTS packages are currently installed on the system:

```
# pkginfo -c sunvts
```

- If the system does not display anything, then *no* SunVTS packages are installed and you can proceed with the installation.

- If you see any messages such as the following message, then SunVTS is installed and you should remove the existing SunVTS software before proceeding. See “Removing SunVTS” on page 2-16.

```
system      SUNWvts      SunVTS
```

### 3. Load the Software Supplement for the Solaris Operating Environment CD (the Supplement CD) into the CD-ROM drive.

Volume Manager automatically mounts the CD.

### 4. Install SunVTS using `pkgadd` as shown in one of the examples below.

- The following command installs SunVTS in the default directory (`/opt`):

```
# pkgadd -d /cdrom/cdrom0/SunVTS_5.1/Product SUNWvts
```

- The following command prompts you to enter the directory where SunVTS will reside instead of using the default (`/opt`) directory.

```
# pkgadd -a none -d /cdrom/cdrom0/SunVTS_5.1/Product SUNWvts
```

---

**Note** – If you install SunVTS in a directory other than the default (`/opt`) directory, you must set the `VTS_PM_PATH` variable before you use the SunVTS CDE interface. See “To Perform Additional Setup When SunVTS Is Installed in a Directory Other Than `/opt`” on page 2-6.

---

### 5. Answer the installation questions.

You are asked if you want to enable the Kerberos-based, Sun Enterprise Authentication Mechanism (SEAM) security which provides the highest level of SunVTS security. You should only enable this if you have the SEAM software installed and the SEAM server and clients configured in your networked environment. Refer to “SunVTS Security” on page 2-7. You can always answer no to this question, even when SEAM is configured in your environment.

If you choose to run SunVTS with SEAM security, use the following SEAM assignments:

- Principal—set to `sunvts`
- Complete Service Name—set to `sunvts@host`, where *host* is the fully qualified domain name of the host where the SunVTS kernel is running.

---

**Note** – You may see installation messages that inform you of other package requirements. These notifications do *not* prevent the successful installation or execution of SunVTS. The following steps explain how to install these other packages.

---

**6. Install the SunVTS supporting packages that are appropriate for your configuration. (See “SunVTS Packages” on page 2-2 for more details.)**

Example:

```
# pkgadd -d /cdrom/cdrom0/SunVTS_5.1/Product SUNWvtsx SUNWvtsmn
```

---

**Note** – Type `-a none` option again if you used it for the installation of `SUNWvts`.

---

---

**Note** – The `SUNWvtsx` package only installs on systems that have the 64-bit Solaris Operating Environment installed.

---

---

**Note** – The `configd` packages (`SUNWeswsa`, `SUNWsyncfd`, `SUNWesnta`, and `SUNWeswga`) are no longer needed, and no longer supplied, for physical mapping support.

---

**7. Verify the presence of the packages:**

```
# pkginfo SUNWvts SUNWvtsx SUNWvtsmn
system    SUNWvts    SunVTS
system    SUNWvtsmn  SunVTS Man Pages
system    SUNWvtsx   64-bit SunVTS
```

## ▼ To Set Up Access to the SunVTS Man Pages

The SunVTS man pages are installed in the `SunVTS_install_dir/man` directory (`/opt/SUNWvts/man` by default). To access them, add this directory to your `MANPATH` shell variable in the initialization file that corresponds to your login shell (usually `.profile` for the Bourne and Korn shells or `.login` for the C shell).

---

**Note** – The steps that follow assume that the default SunVTS installation directory (`/opt`) was used when the SunVTS packages were installed. If this is not the case, adjust the instructions to use the directory where the man pages were installed.

---

1. **Using an editor, add the SunVTS man directory (`/opt/SUNWvts/man` by default) to the `MANPATH` variable in the appropriate initialization file.**

Bourne, Korn Shell example:

```
MANPATH=/usr/share/man:/usr/man:/opt/SUNWvts/man;export MANPATH
```

C shell example:

```
setenv MANPATH /usr/share/man:/usr/man:/opt/SUNWvts/man
```

2. **Have the shell read the modified initialization file by sourcing it (with the `.` [dot] or `source` command) or log out and log back in.**
3. **Verify that the SunVTS man directory is part of the `MANPATH` variable:**

```
# echo $MANPATH
/usr/share/man:/usr/man:/opt/SUNWvts/man
```

---

**Note** – For more information about customizing a user's work environment, shell variables, and initialization files, refer to your Solaris system administration guides.

---

## ▼ To Perform Additional Setup When SunVTS Is Installed in a Directory Other Than `/opt`

When you install SunVTS in a directory other than the default directory (`/opt`), you must set the `VTS_PM_PATH` environment variable before you can use the SunVTS CDE user interface. This variable is used to locate graphic elements in the SunVTS CDE interface.

---

**Note** – The `VTS_PM_PATH` variable is not required if SunVTS is installed in the default directory (`/opt`).

---



1. Using an editor, open the appropriate initialization file such as `.profile` (for the Bourne or Korn shells) or `.login` (for the C shell).

2. Add the `VTS_PM_PATH` variable as shown:

Bourne or Korn Shell example:

```
VTS_PM_PATH=your_base_install_dir/SUNWvts/bin/pm;export VTS_PM_PATH
```

C shell example:

```
setenv VTS_PM_PATH your_base_install_dir/SUNWvts/bin/pm
```

3. Have the shell read the modified initialization file by sourcing it (with the `.` [dot] or `source` command) or log out and log back in.

---

## 2.4 SunVTS Security

SunVTS has two security mechanisms that you can choose from:

- Basic Security—a local file is maintained that lists the valid users, groups, and hosts that are permitted to use SunVTS. This level of security does not provide secure network authentication and should not be used in an environment where network security is an issue.
- SEAM Security—uses the Sun Enterprise Authentication Mechanism™ (SEAM) to provide secure user authentication, data integrity and privacy, for transactions over networks using a Kerberos V5 technology.

The SunVTS installation process prompts you to specify which security mechanism you want to use. You must use one or the other, and the SEAM security implementation is the default if you press the return key through the installation questions.

### 2.4.1 Basic Security

The SunVTS user interface (`vtsui` and `vtstty`) must connect to the SunVTS kernel (`vtsk`) *before* it can be used to control testing. The SunVTS kernel selectively accepts “connect to” requests from the SunVTS interface based on entries in the `SunVTS_install_dir/bin/.sunvts_sec` file. Connection permission is governed by three categories in this file as follows:

- **HOSTS**—if the requesting user is on a host that belongs to the HOSTS category, then this request is granted without any authentication.
- **GROUPS**—if the requesting user is a member in a group that is a member of GROUPS category, then the user interface will prompt the user for a password. The SunVTS kernel compares this password against the password database on the system under test (SUT). If the password does not match, or if the user is not on the list, then the connection is rejected.
- **USERS**—if the requesting user is a member of the USER category, then the user interface will prompt the user for a password. The SunVTS kernel compares this password against the password database on the SUT. If the password does not match, or the user is not on the list, then the connection is rejected.

A plus (+) entry in one of these categories means *all* hosts, groups, or users, are trusted.

The user password needed for authentication is the same password used to log in to the system under test.

The check for connection permission starts with the HOSTS category, then the GROUP category, and finally, the USERS category. A connection is granted as soon as the connection request matches an entry.

If a security file entry is invalid or if there is no entry in the file, all access except root is denied on the local machine. However, you can correct an entry in this file even while the SunVTS kernel is running.

When you specify the `-e` option while starting the SunVTS kernel, the kernel accepts “connect to” requests from any host, regardless of the entries in the `.sunvts_sec` file.

---

**Note** – As of SunVTS 3.1, the `.sunvts_sec` file, by default, is configured for root on the system under test. All other “connect to” requests are rejected.

---

---

**Note** – The `.sunvts_sec` file is bypassed if you enable the SEAM security.

---

The following shows the contents of the default `.sunvts_sec` file.

### Code Example of the Security File (.sunvts\_sec):

```
#This file should be <SunVTS 5.1 install
directory>/bin/.sunvts_sec
#
#Any line beginning with a # is a comment line
#
# Trusted Hosts entry
# One hostname per line.
# A "+" entry on a line indicates that ALL hosts are Trusted Hosts.
# No password authentication is done.
# The line with the label HOSTS: is required to have the list of
hosts
#
HOSTS:
#+
#host1
#host2
#
# Trusted Groups entry
# One groupname per line.
# A "+" entry on a line indicates that ALL groups are Trusted
Groups.
# User password authentication is done.
# The line with the label GROUPS: is required to have the list of
groups
#
GROUPS:
#group1
#
# Trusted Users entry
# One username per line.
# A "+" entry on a line indicates that ALL users are Trusted Users.
# User password authentication is done.
# The line with the label USERS: is required to have the list of
users.
USERS:
root
#user1
#user2
```

## 2.4.2 SEAM Security

To use SEAM-based security with SunVTS, you must have the following:

- The complete SEAM 1.0.1 client/server application must be installed and running in your network environment.
- The systems for which you install SunVTS must have, at minimum, the SEAM 1.0.1 client software installed.
- Select the SEAM-based security (Kerberos V5) choice during the SunVTS installation.

---

**Note** – Refer to the following documents for more information on SEAM:

- *Sun Enterprise Administration Mechanism 1.0.1 Guide*
- *SEAM 1.0.1 Installation and Release Notes*

These documents are part of the Sun Enterprise Authentication Mechanism 1.0.1 AnswerBook Collection, and available at <http://docs.sun.com>.

The SEAM software is part of the Solaris release.

---

The SunVTS SEAM security system is based on Kerberos V5 technology, which revolves around the concept of a *ticket*. A ticket is a set of electronic information that serves as identification for a user or a service. When you connect to another host through SunVTS, you transparently send a request for a ticket to a Key Distribution Center (KDC), which accesses a database to authenticate your identity. The KDC returns a ticket granting you permission to access the other machine.

“Transparently” means that you do not need to explicitly request a ticket; it happens in the background as part of the remote connection. No user password is transmitted in the network. Only the authenticated client can get a ticket for a specific service; another client cannot gain access under an assumed identity.

If you choose to run SunVTS with SEAM security, use the following SEAM assignments:

- Principal—set to `sunvts`
- Complete Service Name—set to `sunvts@host`, where *host* is the fully qualified domain name of the host where the SunVTS kernel is running.

## 2.4.3 Controlling SunVTS Security

### ▼ To Control SunVTS Security mode at Installation Time

The best time to establish the SunVTS security mode is during the SunVTS installation.

#### 1. Decide which level of security you want to use with SunVTS.

If you decide to use the SEAM security (highest level of security), make sure that your system is running SEAM.

## 2. Install SunVTS as described in “Installing SunVTS” on page 2-3.

The installation program asks you if you want SEAM security enabled. Answer accordingly:

- **Yes** (the default)—The Kerberos SEAM security is enabled for SunVTS. No additional action is required to administer SunVTS security. SunVTS uses the authentication as defined in the SEAM software configuration in your network environment to grant and deny access to SunVTS. Do NOT select this security scheme if you do not have the SEAM software installed and configured in your network environment.
- **No**—The basic security file is used, and SEAM is not enabled. When the installation is complete, you can access SunVTS as superuser on the system under test, or modify the `.sunvts_sec` file to authorize other users.

## ▼ To Switch SunVTS Security After Installation

If you need to switch the security from SEAM to basic, or vice-versa, after you have installed SunVTS, follow these steps:

1. **Become superuser.**
2. **Make sure that SunVTS is not started.**
3. **Change directories to the SunVTS binary directory:**

```
# cd /opt/SUNWvts/bin
```

---

**Note** – If SunVTS is installed in a directory other than `/opt`, adjust your references accordingly.

---

4. **With an editor, open the `.sunvts_sec_gss` file.**

This file contains one line that is appended with one of the following:

- `ON`—indicates SEAM security is enabled.
- `OFF`—indicates SEAM security is disabled, and basic security is used.

5. **Change the `ON` (or `OFF`) value to the opposite value, save the change, and quit the editor.**

---

**Note** – The `ON` and `OFF` values are case sensitive. Make sure you specify them using capital letters.

---

6. **Start SunVTS.**

The security mechanism that you specified is enabled.

---

## 2.5 SunVTS Environmental Variables

Use environmental variables to control certain aspects of SunVTS as described in the following table. Except for the `MANPATH` variable, you only use these variables when you need to alter the default characteristics of SunVTS.

**TABLE 2-2** SunVTS Environmental Variables

Variable	Description
<code>BYPASS_FS_PROBE</code>	Used by <code>disktest</code> when you want to run subtests that require SunVTS to pre-mount all mountable partitions. Refer to the <code>disktest</code> chapter in the <i>SunVTS Test Reference Manual</i> for details.
<code>MANPATH</code>	Append the <code>MANPATH</code> variable with the location of the SunVTS man page location ( <code>/opt/SUNWvts/man</code> by default) so the <code>man</code> command can locate and display the SunVTS man pages. See “To Set Up Access to the SunVTS Man Pages” on page 2-5.
<code>VTS_CMD_HOST</code>	Used by the <code>vts_cmd</code> command to specify the hostname for the SunVTS kernel to connect to. Refer to the <code>vts_cmd</code> man page for details.
<code>VTS_OLD_MSG</code>	As of SunVTS 5.0, this variable is no longer supported. It was used to display test messages in a pre-SunVTS 4.0 format, usually because a script relies on the older format. Now your scripts must accept the current SunVTS message format.
<code>VTS_PM_PATH</code>	Only used when SunVTS is not installed in the default directory ( <code>/opt</code> ). Set <code>VTS_PM_PATH</code> to <code>vts_install_dir/SUNWvts/bin/pm</code> for proper operation of the SunVTS CDE UI. See “To Perform Additional Setup When SunVTS Is Installed in a Directory Other Than <code>/opt</code> ” on page 2-6.

---

## 2.6 Additional Instructions for Localized Environments

SunVTS software is available in English, and is Internationalization (I18N) compliant, which means it is structured in a way that a user who is familiar with internationalization can take advantage of this feature to run SunVTS in a localized environment.

In a localized environment, you can run SunVTS in English *or* with localized fonts. Perform one of the two following procedures based on how you want SunVTS to run.

## ▼ To Run SunVTS in English, in a Non-English Environment

Set the `LANG` variable to English before you start SunVTS. The following is an example using the C shell:

1. **Set the `LANG` variable:**

```
# setenv LANG C
```

## ▼ To Set Up the GUI resource File for Localized Environments

1. **As superuser, create a directory as follows:**

```
# mkdir -p /opt/SUNWvts/lib/locale/LANG/app-defaults
```

Where `LANG` is the language code (for example, `ja` for Japan EUC, or `fr` for French) for the language that you are using.

2. **Copy the SunVTS Xresource (`Vtsui`) file into the directory you just created:**

```
# cp /opt/SUNWvts/lib/Vtsui /opt/SUNWvts/lib/locale/LANG/app-  
defaults
```

---

**Note** – The example above is based on installing SunVTS in the default (`/opt`) directory. If SunVTS is installed in different directory, adjust the pathnames accordingly.

---

3. **Customize the font definitions in the `Vtsui` file to accommodate font specifications required by your localized environment.**

---

## 2.7 Adding a Custom Test

Developers can add their own custom tests to the SunVTS environment. This book does not describe custom test development, but does list the tasks necessary to add a custom test into the SunVTS environment.

### ▼ To Add a Custom Test

**1. Copy your custom test binary to the SunVTS bin directory based on the 32-bit or 64-bit functionality of your binary test:**

- 32-bit: `/opt/SUNWvts/bin`
- 64-bit: `/opt/SUNWvts/bin/sparcv9`

**2. Modify one of the following `.customtest` files based on the 32-bit or 64-bit functionality of your binary test:**

- 32-bit: `/opt/SUNWvts/bin/.customtest`
- 64-bit: `/opt/SUNWvts/bin/sparcv9/.customtest`

The format of the `.customtest` file is described in “The `.customtest` File Format” on page 2-14.

**3. Restart SunVTS or reprobe the system.**

When invoked, SunVTS displays the custom test in the SunVTS user interface.

### 2.7.1 The `.customtest` File Format

The `.customtest` file defines the test options and the default option values for your custom test. The tester can change these options using the option dialog boxes through the SunVTS user interface. However, the Reset button returns the options to the default settings as defined in the `.customtest` file.

Each line in this file is made up of two or more fields that are separated by a semicolon where:

- The first field is the label or device name (mandatory field).
- The second field is the test name (mandatory field).
- The third field is an option line (optional field). If used, this field must be in the format specified.



- The fourth field is used if the test is scalable. If used, append the keyword *SCA* to this field.

Examples:

- To add a test with no options:

```
% your_label_name;your_test_name
```

- To add the scalability option, append the keyword *SCA*:

```
% your_label_name;your_test_name;SCA
```

- To custom build an option menu, add an option specification:

```
% Option_Name<Option_Type | Value | Default_Value | Command_Line_Option>
```

- To specify more than one option, separate each option by a comma:

```
% label_name;test_name;Numeric<NUMERIC | 0,100 | 50 | numeric>,
Exc_Choice<EXC_CHOICE | Top, Middle, Bottom | Middle | exc_choice> ,
Inc_Choice
<INC_CHOICE | Left, Center, Right | Left+Center+Right | inc_choice> ,
Toggle<TOGGLE | This, That | This | toggle> ,
Text<TEXT | 20 | Type_Here | text> , Slider<SLIDEBAR | 0, 10 | 5 | slider> ,
Errors<CYCLE | Yes, No | No | errors> ,
Cycle<CYCLE | First, Second, Third | First | cycle> ;SCA
```

SunVTS invokes the above test as follows:

```
% ./test_name -s[vq..] [-i n]
-o dev=user[0,1..],Command_Line_Option=Value...
```

You cannot use the `.customtest` file when a test has a probe attached. You must ensure that the binaries are compatible with the version of the Solaris kernel on which SunVTS is currently running.

---

**Note** – If `.customtest` is renamed as `.customtest-group`, all of the associated tests will appear under the specified *group*.

---

---

## 2.8 Removing SunVTS

It is necessary to remove SunVTS before installing a new version. The following instructions describe how to remove SunVTS using the `pkgrm` command.

Use the same tool or utility for installation and removal of the SunVTS software. If you use `pkgadd` for installation, use `pkgrm` to uninstall; if you use Web Start for installation, use the product registry to uninstall.



---

**Caution** – If you used Web Start to install the SunVTS packages, make sure to use the product registry to remove the packages. Using `pkgrm` to remove packages that were installed with Web Start could cause the package database and the product registry to be out of sync.

---

### ▼ To Remove SunVTS With the `pkgrm` Command

1. Log in to the system and become superuser.

```
% su
```

2. Remove the packages with the `pkgrm` command:

```
# pkgrm SUNWvtsx SUNWvtsmn SUNWvtsol SUNWvts
```

---

**Note** – As of SunVTS 5.0, the `SUNWvtsol` package is obsolete. You should remove it if it is installed on the system for which you plan to install SunVTS 5.1.

---

Answer `y` (yes) to the removal questions.

You should see Removal of `package_name` was successful.

## Starting SunVTS

---

This chapter contains information that describes:

- “SunVTS Runtime Considerations” on page 3-1
- “Requirements for Starting SunVTS” on page 3-2
- “Overview of SunVTS Procedures” on page 3-3
- “Preparing Devices for Testing” on page 3-4
- “Starting SunVTS on a Local System” on page 3-5
- “Starting SunVTS on a Remote System” on page 3-9
- “Quitting SunVTS” on page 3-13
- “Displaying Devices With the vtsprobe Utility” on page 3-13

---

### 3.1 SunVTS Runtime Considerations

Before you start SunVTS, consider the following runtime issues:

- **System load**—Before you run SunVTS on your system you should consider the added system activity created by SunVTS and consider its effect on users. Depending on the mode that you select, testing can add significantly to your system load or be quite minimal. Added load is desirable when using SunVTS for system stress testing and validation.
- **Other applications**—Some SunVTS tests stress the system resources (such as virtual memory) to the point that other applications might not run.
- **System functionality**—SunVTS is a diagnostic tool. It only runs when your system is fully booted to run level 3 (multi-user level). SunVTS will not run from single-user (maintenance mode) nor from run level 0 (monitor mode).

- Superuser access—For security reasons, only superuser is permitted to run SunVTS unless the SunVTS security file is modified when you use the basic SunVTS security scheme.
- Graphics Tests—Running tests on frame buffers has two key considerations: You must not run any other application or screen saver program that uses the frame buffer. The frame buffer graphic tests print test patterns on the display. This temporarily prevents you from working in your window environment.
- Swap Space—The amount of swap space required for testing varies widely with individual hardware and software configurations. If an adequate amount of swap space is not available on your machine, a window pops up and displays the amount of additional swap space needed.
- Latest SunVTS news—Refer to the `/opt/SUNWvts/README` file for the latest SunVTS updates.

---

## 3.2 Requirements for Starting SunVTS

Your system must meet the following requirements to run SunVTS:

- The SunVTS packages must be installed on the systems on which you plan to run SunVTS. Refer to “Installing and Removing SunVTS” on page 2-1 for installation information.
- The system must be booted to the multiuser level.
- To run SunVTS with a graphical user interface (GUI), that GUI must be installed (CDE). Otherwise, start SunVTS with the TTY interface.

---

**Note** – Although it is not a requirement, it is good practice to prepare devices like tape drives, CD-ROM drives, and diskette drives by installing the proper media before starting SunVTS. This practice assures the most reliable device-probe results. See “Preparing Devices for Testing” on page 3-4 for more information.

---

---

## 3.3 Overview of SunVTS Procedures

The following table describes the process for testing your system with SunVTS. The table also provides a brief description, and refers you to the location in the document for more detailed information.

TABLE 3-1 SunVTS Procedures

Step	Procedure	Description
1	Become superuser	Log in as root or use <code>su</code> to become root (superuser). Note: To access SunVTS the user or host must be granted access through one of the SunVTS security mechanisms. See “SunVTS Security” on page 2-7.
2	Determine the current state of the system you plan to test	Your system is considered <i>online</i> if other applications are running and the system is operating in a production capacity. If the system is online, you should stop the applications and make sure the system remains offline for the duration of your testing. See “SunVTS Runtime Considerations” on page 3-1.
3	Prepare devices for testing	Load the required test media and loopback connectors. Test media is required when testing your tape, CD-ROM, and diskette drives in Functional mode. Communication port tests require loopback connectors. Refer to “Preparing Devices for Testing” on page 3-4.
4	Start your desired graphical environment (optional)	SunVTS runs in the CDE window environment, or in a non-graphical window environment (in TTY mode).
5	Start SunVTS	There are many ways to start the SunVTS application. Refer to the following sections: <ul style="list-style-type: none"><li>• “Starting SunVTS on a Local System” on page 3-5</li><li>• “Starting SunVTS on a Remote System” on page 3-9</li><li>• “Displaying Devices With the <code>vtsp</code> Utility” on page 3-13</li></ul>
6	Set up SunVTS with your desired test session configuration	The method for doing this depends on the SunVTS interface that you are using: <ul style="list-style-type: none"><li>• “Using the SunVTS CDE User Interface” on page 4-1</li><li>• “Using the SunVTS TTY User Interface” on page 5-1</li></ul>

TABLE 3-1 SunVTS Procedures (Continued)

Step	Procedure	Description
7	Run the tests	To run the tests, select the <code>start</code> button in the SunVTS interface. The testing process may take as little as one second, or run indefinitely, based on how SunVTS is configured.
8	Review the test results	Test results are displayed real-time in the SunVTS interface and logged in three log files. You can also configure SunVTS to notify you by email.
9	Quit SunVTS	Refer to “Quitting SunVTS” on page 3-13 for more details.

## 3.4 Preparing Devices for Testing

Several tests require the installation of media or loopback connectors. Install such requirements *before* the SunVTS kernel probes for devices (when SunVTS is started, or when the SunVTS `reprobe` command is run) to allow the SunVTS kernel to properly identify each device.

You must install media in these devices if you plan to test them in Functional Mode:

- Tape drive—Load a *scratch* tape in the tape drive that you plan to test. The tape test has a Read-only mode, but use a scratch tape to avoid inadvertently writing over data.
- CD-ROM and DVD drives—Load appropriate media into the drive. It can be an audio or data CD.
- Diskette drive—Load a *scratch* diskette in the drive. The diskette test has a Read-only mode, but use a scratch tape to avoid inadvertently writing over data.
- SmartCard Reader—Allocate a *scratch* smart card for testing purposes because the SmartCard test writes on the card.
- Communication ports—Many of these tests require a loopback connector attached to the port. Attach any required loopback connectors for the ports you plan to test. For more information about loopback connectors refer to the *SunVTS Test Reference Manual*.
- Printer ports—Requires connection to a printer.

---

**Note** – Any device that requires your intervention before the test can run (such as loading media for tape, CD-ROM, and diskette tests), also requires that you notify SunVTS that you have performed this action by selecting the Intervention checkbox. You cannot select any intervention mode tests until you select this checkbox.

---

---

## 3.5 Starting SunVTS on a Local System

This section describes the common ways to start SunVTS:

- “Using the `sunvts` Command” on page 3-5
- “Starting the SunVTS Kernel and Interface Separately” on page 3-7
- “Starting SunVTS to Test an Online System” on page 3-9

### 3.5.1 Using the `sunvts` Command

The typical way to start SunVTS is to use the `sunvts` command. This command starts the SunVTS kernel, either the 32-bit or 64-bit version, and one of the SunVTS user interfaces, based on your system environment.

The SunVTS UI that starts is determined by the following criteria:

- When the CDE window manager (`DTWM`) is running, the SunVTS CDE UI is started (`vtsui`).
- If no window manager is running, then the SunVTS TTY UI is started (`vtstty`).

---

**Note** – The `sunvts` command starts SunVTS assuming that the system under test is in an offline state. All other system applications should be stopped.

---

#### ▼ To Start SunVTS with the `sunvts` command

1. Become superuser.
2. Run the `sunvts` command:

```
# /opt/SUNWvts/bin/sunvts
```

---

**Note** – If you receive an error message such as:

```
connection to ":0.0" refused by server
Xlib: Client is not authorized to connect to Server
Error: Can't open display: :0.0
```

You may need to grant `xhost` permissions by typing: `xhost display_hostname`

---

3. Refer to one of the following chapters according to the SunVTS UI you are using:
  - “Using the SunVTS CDE User Interface” on page 4-1

- “Using the SunVTS TTY User Interface” on page 5-1

## ▼ To Use the `sunvts` Command With Options

You can use the `sunvts` command with a variety of options (as shown in the following table) to control how SunVTS is started.

1. For example, the following command starts SunVTS with the TTY UI no matter what window environment you are using:

```
# /opt/SUNWvts/bin/sunvts -t
```

**TABLE 3-2** The `sunvts` Command Syntax

```
/opt/SUNWvts/bin/sunvts [-epqstv] [-o options_file] [-f logfile_directory] [-h hostname] -
display local_hostname:0
```

Argument	Description
<code>-e</code>	Disables the connection permission checking feature
<code>-p</code>	Starts the SunVTS kernel, but does not probe the test system's devices
<code>-q</code>	Automatically quits both the SunVTS kernel and the user interface when testing stops
<code>-s</code>	Automatically starts testing a selected group of tests; the flag must be used with the <code>-o options_file</code> flag
<code>-t</code>	Starts <code>vtstty</code> , a TTY-based program. If you do not specify this option, and the system is running CDE, <code>sunvts</code> starts <code>vtsui</code> .
<code>-v</code>	Displays version information from the SunVTS kernel and UI
<code>-o options_file</code>	Starts the SunVTS kernel with the test options loaded from the <code>options_file</code> ; these options are saved by the user interface and are stored in the <code>/var/opt/SUNWvts/options</code> directory
<code>-f logfile_directory</code>	Specifies an alternative logfile directory other than the default directory <code>/var/opt/SUNWvts/logs</code>
<code>-h hostname</code>	Starts the user interface ( <code>vtsui</code> or <code>vtstty</code> ) on the local machine and tries to connect to the SunVTS kernel ( <code>vtsk</code> ) of the specified host machine. If <code>hostname</code> is the same hostname of the machine on which the tests are being run, <code>sunvts</code> starts the SunVTS kernel ( <code>vtsk</code> ). If <code>vtsk</code> is already running on the test system, the <code>sunvts</code> command ignores the <code>-o</code> , <code>-f</code> , <code>-q</code> , <code>-p</code> , and <code>-s</code> options
<code>-display local_hostname:0</code>	When running SunVTS through a remote login, this option starts the SunVTS kernel on the remote system, and the user interface is displayed on the local machine designated as <code>local_hostname:0</code>



## 3.5.2 Starting the SunVTS Kernel and Interface Separately

The `sunvts` command starts both the SunVTS kernel and the user interface that is appropriate for your system. However, there are commands that you can use to start just the SunVTS kernel, or to specify a specific user interface.

### ▼ To Start the SunVTS Kernel (`vtstk`)

1. **The `vtstk` command starts the SunVTS kernel. You can use command-line options (see the following table below) to control the behavior of this command.**
  - For systems running in the 32-bit Solaris environment, use the following command:

```
# /opt/SUNWvts/bin/vtsk
```

- For systems running in the 64-bit Solaris environment, use the following command:

```
# /opt/SUNWvts/bin/sparcv9/vtsk
```

The `vtsk` command-line syntax and options are listed below:

**TABLE 3-3** The `vtsk` Syntax

```
/opt/SUNWvts/bin/vtsk [-epqsv] [-o options_file] [-f logfile_directory]
```

Argument	Description
<code>-e</code>	Disables the connection permission checking feature
<code>-p</code>	Starts the SunVTS kernel, but does not probe test system devices
<code>-q</code>	Quits both the SunVTS kernel and the user interface when testing is complete
<code>-s</code>	Starts testing a selected group of tests; this flag must be used with the <code>-o options_file</code> flag
<code>-v</code>	Displays only the version information from the SunVTS kernel, <code>vtsk</code> ; this option does not start the <code>vtsk</code> daemon
<code>-o options_file</code>	Starts the SunVTS kernel with the test options saved in the <code>options_file</code> ; these options are saved and stored in the <code>/var/opt/SUNWvts/options</code> directory
<code>-f logfile_directory</code>	Specifies an alternative to the default logfile directory ( <code>/var/opt/SUNWvts/logs</code> )

## ▼ To Start a Specific SunVTS User Interface

In most cases, when you use the `sunvts` command, the appropriate SunVTS user interface is started.

### 1. For situations when you want to start a specific user interface without starting the SunVTS kernel, use one of the following:

- To start the CDE interface:

```
# /opt/SUNWvts/bin/vtsui
```

- To start the TTY interface:

```
# /opt/SUNWvts/bin/vtstty
```

In all cases, the user interface attempts to connect to the SunVTS kernel.

### 3.5.3 Starting SunVTS to Test an Online System

As of SunVTS 4.3, the SunVTS online testing capability that was initiated using the `vtsui.online` command is no longer available. The `SUNWodu` package that provides this online testing functionality is no longer provided.

Online Diagnostic testing of Sun systems is now available through the Sun Management Center software using the Sun Hardware Diagnostic Suite add-on software. See <http://www.sun.com/sunmanagementcenter> for details.

Before you start a SunVTS test session stop all user applications.

---

## 3.6 Starting SunVTS on a Remote System

You can start, view, and control SunVTS testing sessions over a network. There are several ways to do it, but the desired effect is to have SunVTS display the SunVTS user interface on the *local system* (the system that you are on), while the SunVTS kernel is running and testing a *remote system* (connected via Ethernet, modem lines, and the like).

The following procedures are described in this section:

- “To Connect the User Interface to a Remote System” on page 3-10
- “To Run SunVTS Through a Remote Login” on page 3-11
- “To Run SunVTS Through telnet or tip (TTY Interface)” on page 3-11

### 3.6.1 Requirements

In addition to the normal SunVTS requirements (refer to “Requirements for Starting SunVTS” on page 3-2), the following requirements must be met to run SunVTS on a remote system:

- The user or host making the remote connection must be an authorized SunVTS user, either through the `.sunvts_sec` file (basic security), or through the SEAM security scheme. See “SunVTS Security” on page 2-7.
- The same version of SunVTS must be installed on both the local and remote systems.

## ▼ To Connect the User Interface to a Remote System

### 1. Type the following command:

```
# /opt/SUNWvts/bin/sunvts -h remote_hostname
```

Replace *remote\_hostname* with the hostname or IP address of the remote system.

The `sunvts` command starts the SunVTS kernel (`vtstk`) on the remote system, and starts the SunVTS user interface on the local system. The user interface connects to the kernel and displays the test session of the remote system. Refer to FIGURE 3-1 below.

The term *system under test (SUT)* refers to the system running the SunVTS kernel, which is the remote system in this example.

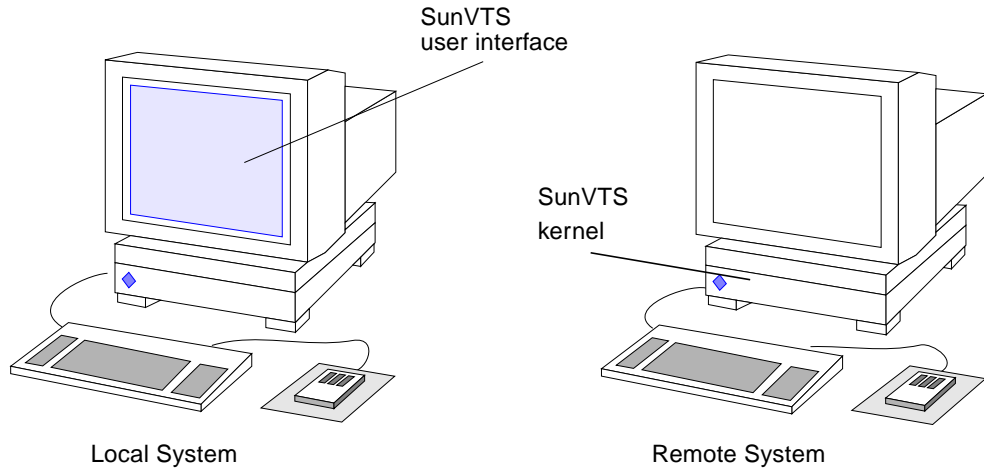


FIGURE 3-1 SunVTS CDE Main Window

---

**Note** – If the user interface is already running on your local system, you can use the “Connect to” button to connect to a remote machine SunVTS kernel, provided that the SunVTS security is set up properly.

---

**2. Configure SunVTS for the test session and start the tests as described in the following sections:**

- “Using the SunVTS CDE User Interface” on page 4-1
- “Using the SunVTS TTY User Interface” on page 5-1

## ▼ To Run SunVTS Through a Remote Login

**1. Use the `xhost` command to allow the remote system to display on your local system.**

```
% /usr/openwin/bin/xhost + remote_hostname
```

Replace *remote\_hostname* with the name of the remote system.

**2. Log in to the remote system as superuser.**

Use a command such as `rlogin`.

**3. Start SunVTS:**

```
# /opt/SUNWvts/bin/sunvts -display local_hostname:0
```

Replace *local\_hostname* with the name of the local system.

The SunVTS kernel starts on the remote system and the user interface displays on your system.

**4. Configure SunVTS for the test session and start the tests.**

- “Using the SunVTS CDE User Interface” on page 4-1
- “Using the SunVTS TTY User Interface” on page 5-1

## ▼ To Run SunVTS Through `telnet` or `tip` (TTY Interface)

You can run SunVTS on a remote system with the TTY interface through a `telnet` or `tip` session.

You need to set the correct terminal type and number of columns and rows *before* starting the interface. The steps below describe this process.

**1. Use the `echo` command to display the value of the `TERM` variable:**

In this example, the `TERM` variable is a Korn or Bourne shell variable and the *value* is `sun-cmd`. Your display device may be something else like a Wyse, TeleVideo, or other kind of terminal. In that case your `TERM` value will be something different.

```
$ echo $TERM
sun-cmd
```

**2. Use the `stty` command to display the settings of your terminal:**

```
$ stty
speed 9600 baud; -parity hupcl
rows = 60; columns = 80; ypixels = 780; xpixels = 568;
swtch = <undef>;
brkint -inpck -istrip icrnl -ixany imaxbel onlcr
echo echoe echok echoctl echoke iexten
```

---

**Note** – You must have a minimum of 80 columns and 24 rows to run the SunVTS TTY interface.

---

**3. Write down the values of your `TERM` variable and `rows` and `columns` settings.**

You will need these values later.

**4. Connect to the remote system using either the `telnet` or `tip` commands.**

Refer to the `telnet(1)` and `tip(1)` man pages for more information about these commands.

**5. Become superuser on the remote system.**

**6. Identify your terminal type and settings in the `telnet` (or `tip`) session window:**

```
# TERM=sun-cmd
# stty rows 60
# stty columns 80
```

**7. Start SunVTS with the TTY interface:**

```
# /opt/SUNWvts/bin/sunvts -t
```

8. **Configure SunVTS for the test session and start the tests.**  
Refer to “Using the SunVTS TTY User Interface” on page 5-1.

---

## 3.7 Quitting SunVTS

### ▼ To Quit SunVTS

1. **If SunVTS is testing, stop the test session before quitting SunVTS.**
2. **To quit SunVTS, access the Quit SunVTS submenu and choose from one of the following:**
  - `vt s kernel and ui`—terminates the user interface and the SunVTS kernel
  - `vt s kernel`—terminates only the SunVTS kernel
  - `vt s ui`—terminates only the user interface
  - **Quit Button**—terminates only the user interface

---

## 3.8 Displaying Devices With the `vt sprobe` Utility

The `vt sprobe` command lists all of the system testable devices, associated configuration information, and corresponding hardware tests.

The SunVTS kernel must be running on the test machine for the `vt sprobe` command to work. See “Starting the SunVTS Kernel and Interface Separately” on page 3-7.

Any user or remote host that wishes to run `vt sprobe` must be listed in the `.sunvts_sec` file. By default, root on the local host is listed.

### ▼ To Display the Devices for a Local System

1. **Type the following command:**

**CODE EXAMPLE 3-1** vtsprobe Example

```
example% /opt/SUNWvts/bin/vtsprobe

Processor(s)
  fpu(fputest)
    Architecture: sparc
    Type: TI TMS390Z50 SuperSPARC chip
  system(systemst)
    System Configuration: sun4m SPARCstation 10 (1 X 390Z50)
    System clock frequency: 40 MHz
    SBUS clock frequency: 20 MHz

Memory
  kmem(vmem)
    Amount: 233580KB
  mem(pmем)
    Physical Memory size:48 Mb

Network
  isdn0(isdntest)
  le0(nettest)
    Host_Name: example
    Host Address: 131.155.56.122
    Host ID: 12347f61
    Domain Name: widget.com

SCSI-Devices(esp0)
  c0t0d0(rawtest)
    Logical Name: c0t0d0
    Capacity: 510.23MB
    Controller: esp0
  c0t0d0(fstest)
    Logical Name: c0t0d0
    Controller:esp0
  tape0(tapetest)
    Drive Type: Exabyte EXB-8200 8mm Helical Scan

Comm.Ports
  zs0(sptest)
    term/a & term/b

Graphics
  cgsix0(cg6)
    5000KB required for testing.

OtherDevices
  Controller: Intel 82077
  diskette(fstest)
    Logical Name: diskette
    Controller: Intel 82077
```



**CODE EXAMPLE 3-1** vtsprobe Example (*Continued*)

```
sound0(audio)
  Audio Device Type: DBRI Speakerbox
```

## ▼ To Display Devices of a Remote System

---

**Note** – The user or local host must be listed in the `.sunvts_sec` file on the remote system.

---

1. **Make sure that the SunVTS kernel is running on the remote system.**
2. **Type the following command:**

```
# vtsprobe -h hostname
```

The `vtsprobe` utility connects to the remote machine and displays the hardware devices of the remote machine. The output is displayed on the window in which you invoke `vtsprobe`.



# Using the SunVTS CDE User Interface

---

This chapter describes how to run a test session using the SunVTS CDE UI. The procedures are written in a step-by-step form so you can use them to become familiar with SunVTS using the SunVTS CDE UI. This chapter is divided into the following sections:

- “Getting Started With the SunVTS CDE UI” on page 4-1
- “Using Additional Features With the SunVTS CDE UI” on page 4-18
- “Scheduling Test Sequences With the Deterministic Scheduler” on page 4-30

---

**Note** – This chapter assumes that you have already performed the following procedures:

- Installed SunVTS as described in “Installing and Removing SunVTS” on page 2-1,
  - Started SunVTS as described in “Starting SunVTS” on page 3-1,
  - Prepared your system for testing as described in “Preparing Devices for Testing” on page 3-4.
- 

See Appendix A for descriptions of each SunVTS window and dialog box.

---

## 4.1 Getting Started With the SunVTS CDE UI

This section describes how to use the fundamental features of the SunVTS CDE UI to perform diagnostic testing on a system. For information about using more advanced features, see “Using Additional Features With the SunVTS CDE UI” on page 4-18.

## 4.1.1 The SunVTS CDE UI Main Window

When you start the SunVTS CDE user interface, the SunVTS CDE main window is displayed (FIGURE 4-1). SunVTS is in an idle state with all testing options set to default values.

The procedures in this chapter explain how to use the CDE main window:

- “To Configure SunVTS for a Test Session” on page 4-4
- “To Run, Monitor, and Stop a Test Session” on page 4-13
- “To Review and Reset Test Session Results” on page 4-14

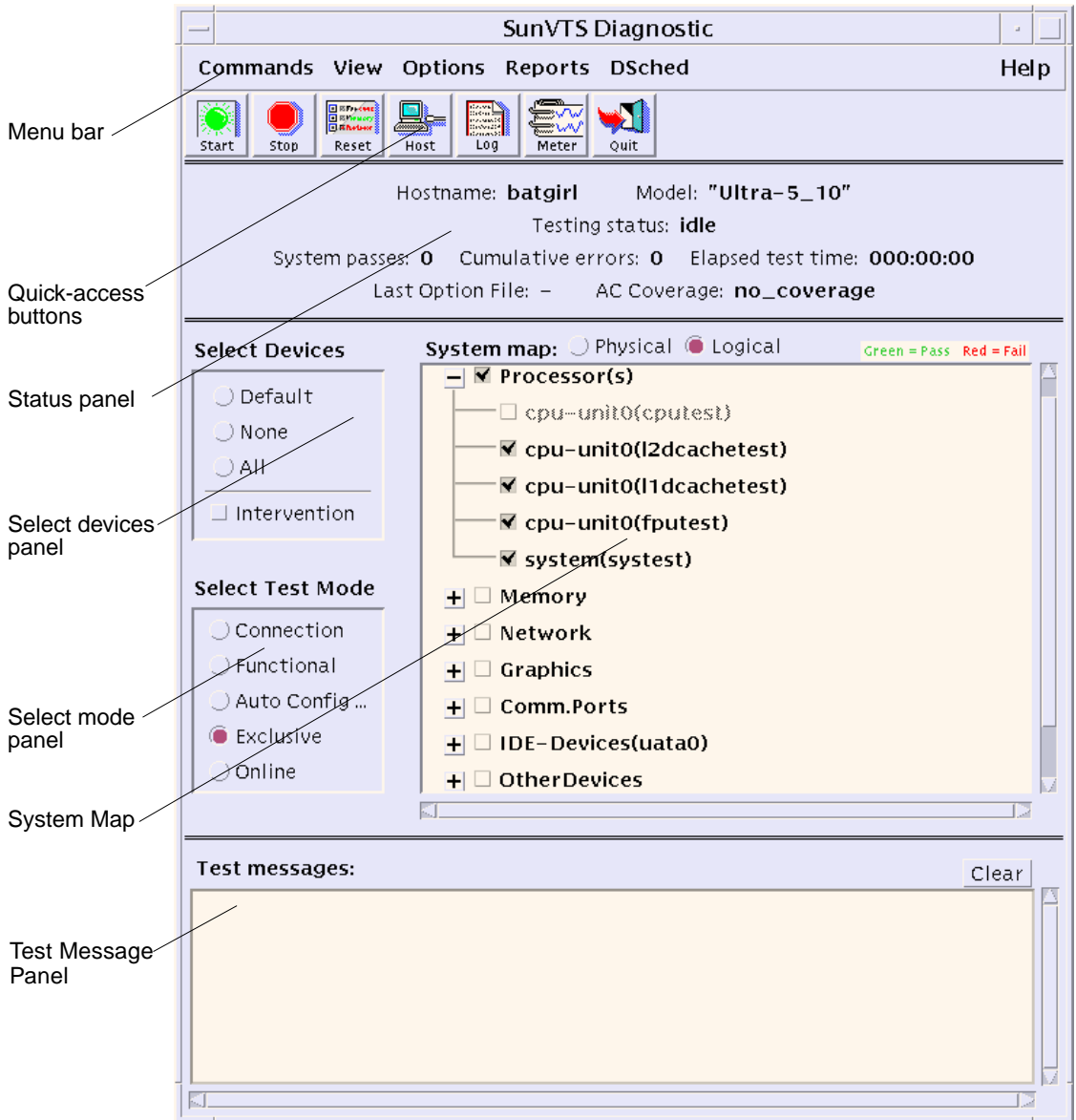


FIGURE 4-1 SunVTS CDE Main Window

---

**Note** – See Appendix A for details about each main window item.

---

## ▼ To Configure SunVTS for a Test Session

### 1. Identify the current state of SunVTS in the Status panel (FIGURE 4-2).

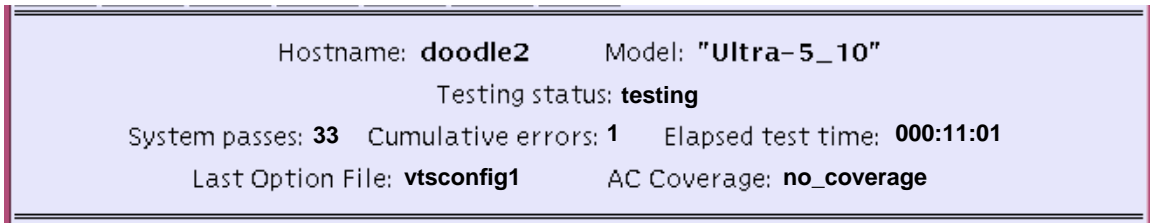


FIGURE 4-2 CDE Status Panel

The following information is displayed in the Status panel:

- Hostname—indicates the name of the system under test
- Model—indicates the model of the system under test
- Testing status—provides the following status of SunVTS:
  - ds\_idle—the deterministic scheduler is invoked, but is not testing
  - ds\_running—the deterministic scheduler is invoked and is running a sequence
  - ds\_suspended—the deterministic scheduler is invoked and a sequence is suspended
  - idle—no testing is taking place
  - replay—a previously recorded test session is displayed
  - stopping—is momentarily displayed when a test session stops
  - suspended—a test session is suspended
  - testing—a test session is running
- System passes—displays the total number of successful system passes (a system pass is when all tests have been run once).
- Cumulative errors—the total number of errors from all tests
- Elapsed test time—indicates the total number of hours, minutes, and seconds that elapsed during testing.
- Last Option File—displays the name of the last option file that was accessed.
- AC Coverage—displays the type of Auto Configuration coverage level that was selected (either `confidence`, `comprehensive`, or `no_coverage` when the AC feature is not in use).

### 2. Select a test mode from the Test Mode panel (FIGURE 4-3):



**FIGURE 4-3** CDE Test Mode Panel

*About test modes:*

- Connection test mode—provides a low-stress, quick testing of the availability and connectivity of selected devices. These tests are non-intrusive, meaning they release the devices after a quick test, and they do not place a heavy load on system activity.
- Functional test mode—provides more robust testing of your system and devices. It will use your system resources for thorough testing and it assumes that no other applications are running.
- Auto Config test mode—opens the Automatic Configuration dialog box which provides access to the Automatic Configuration feature. The Automatic Configuration feature simplifies the test session configuration by assigning a predetermined set of test options. See “Using the Automatic Configuration Feature” on page 4-18 for instructions on using this feature.
- Exclusive test mode—enables users to perform tests that must be performed exclusively, and which allow no other tests or applications to be running concurrently.
- Online test mode—enables users to perform `sunvts` testing while other customer applications are running concurrently.

---

**Note** – As you change test modes, the tests that are selectable in the System Map change.

---

**3. Specify the System Map display mode (logical or physical):**

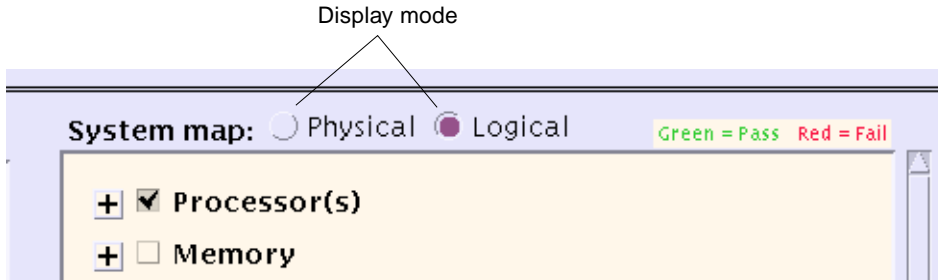


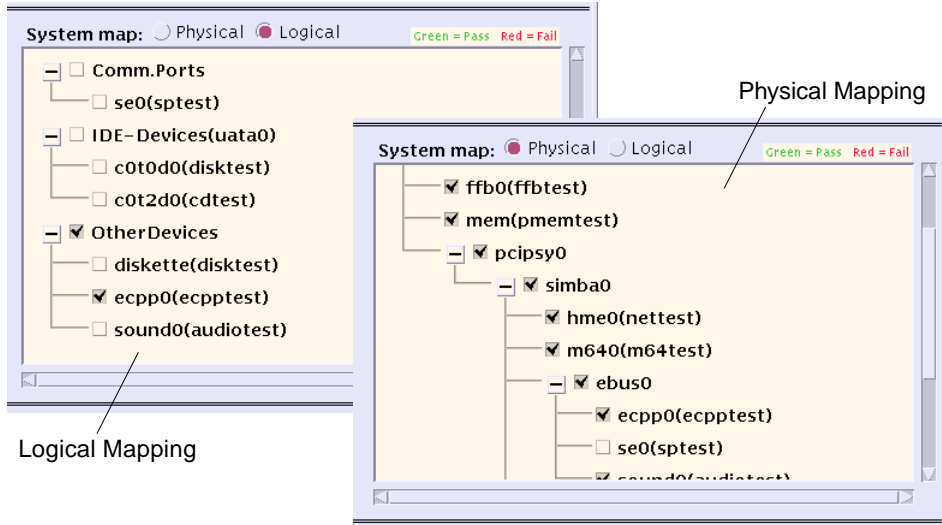
FIGURE 4-4 CDE System Map (Logical View, Collapsed)

*About display modes:*

The devices that SunVTS identified as testable devices on your system are listed in the System Map (FIGURE 4-6). You control the way that SunVTS displays these devices by selecting one of the following:

- **Logical Mapping**—organizes the devices, according to their function, into groups. For example, SCSI disk, SCSI tape, and SCSI CD-ROM drives are placed in the SCSI-Devices group. You can focus your testing on a specific device, device group, or on all of the groups on the system.
- **Physical Mapping**—shows the exact location of each device on the system in relation to how the devices are connected to each other. If you are testing a single-board type system, each device is shown under the system board. Multi-board type systems show each device under the board (for example, board0, board1, and so on) to which the device is attached. For example, if you have multiple disk drives connected to different disk interfaces, each disk would be displayed under the respective interface. From Physical Mapping, you can determine the actual location of each device. When possible, the board number and controller type for the device are also displayed.





**FIGURE 4-5** Logical and Physical Mapping Examples

*How to do it:*

Select the Physical or Logical button.

**4. Expand the System Map to view selectable devices.**

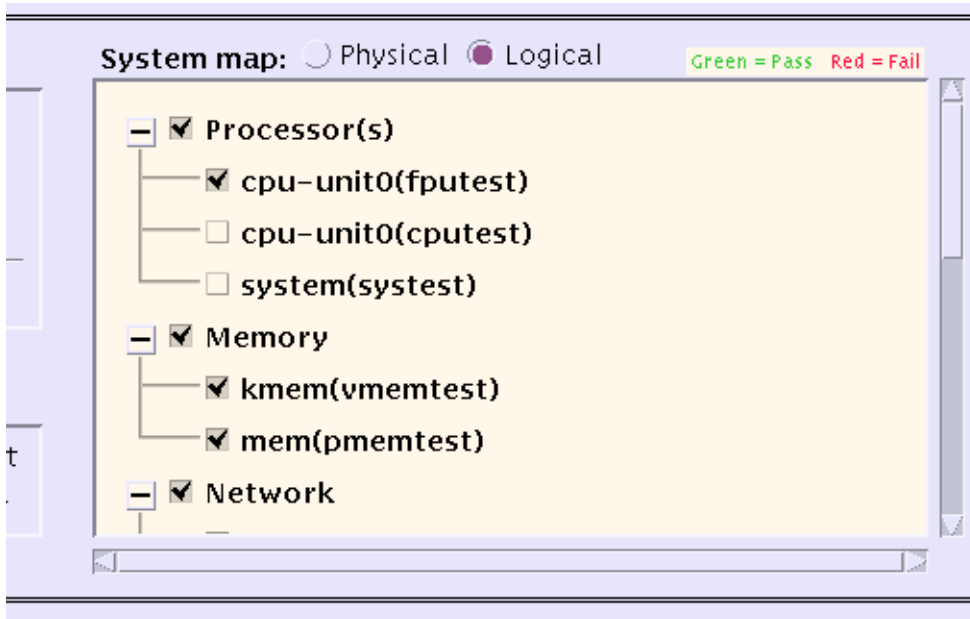


FIGURE 4-6 CDE System Map (Logical View, Expanded)

*About devices in the System Map:*

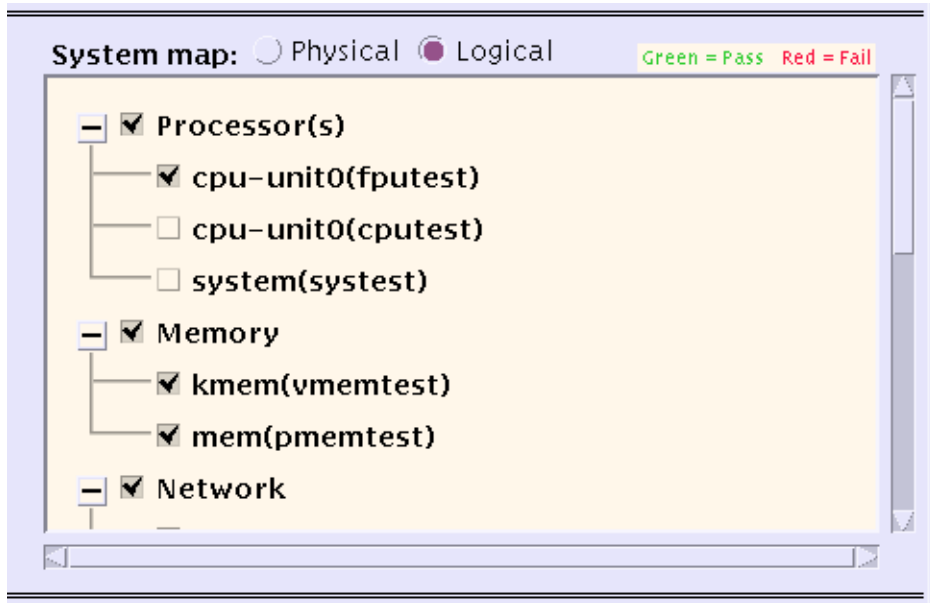
When SunVTS is started, most of the devices in the System Map are *collapsed*, meaning individual devices are grouped together and hidden from view under a device category. A “+” (plus) indicates that the device group is collapsed. A “-” (minus) indicates that the device group is *expanded* so you see each device for a given category.

*How to do it:*

Use the following methods to open the System Map:

- Use the View→Open System Map command from the menu bar to completely expand all the groups.
- Click on the “+” next to a specific group in the System Map to expand an individual group.

**5. Select the devices that you want to test.**



**FIGURE 4-7** Devices Selected in the System Map

*About selecting devices:*

A checkmark next to a device in the System Map indicates that device is selected. Depending on the devices on your system, and the test mode that you select, SunVTS selects certain devices by default.

*How to do it:*

**a. Select your devices using the buttons in the Device Selection panel:**

- Default – selects the SunVTS default selected devices
- None – deselects all devices
- All – selects all devices
- Intervention – required before you can select tests that require special device preparation.

---

**Note** – Any device that requires your intervention before the test can run (such as loading media for tape, CD-ROM, and diskette tests), also requires that you notify SunVTS that you have performed this action by selecting the Intervention checkbox. You cannot select any intervention mode tests until you select this checkbox.

---

**b. Refine the selected devices by clicking on the checkmark box next to a device or device group.**

## 6. Modify the system-level test options (if desired).

### *About test options:*

SunVTS is ready to test each device once you select it. There are, however, testing options that you can modify that alter the way tests run to suit your testing needs. There are three levels where you control test options:

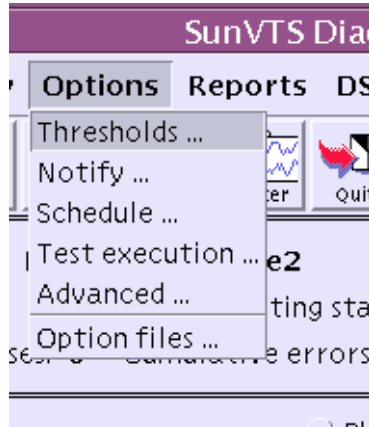
- System-level options—control the testing attributes of *all* devices. This is the highest level, applying all your options globally. When you modify options at this level, the option settings are propagated down to the group-level and device-level options.
- Group-level options—control the testing attributes of all the devices in the group. Settings made at this level propagate down, setting all the options in the levels in the group.
- Device-level options—control the testing of a single, specific device. Some of these options can also be applied to the other like-devices with Apply to All pull-down menus.

The order that you apply the system-level, group-level, and device-level options is important. Start by assigning the system-level options, then assign the group-level options, and finish with the device-level options. Otherwise, if you make lower-level option settings, then higher level option settings, the higher-level settings will propagate down and undo your lower-level settings. An alternative to this approach is to use locks and overrides to protect lower level settings. See “Preserving Test Options with Locks and Overrides” on page 4-23.

---

**Note** – System-level, group-level, and device-level options are only available when Functional Test Mode is selected. Most of these options are *grayed-out* and not available for Connection Mode testing.

---



**FIGURE 4-8** Items From the Options Menu

*How to do it:*

Access the system-level options from the Options menu in the Menu Bar (FIGURE 4-8).

The following dialog boxes provide system-level options that define how the entire test session runs.

- **Thresholds**—provides access to test session limits such as the maximum number of system passes and the maximum test session time.
- **Notify**—provides access to set up testing notification through email.
- **Schedule**—provides access to auto start, single pass, and concurrency options.
- **Test Execution Options**—allows you to define *how* a test session runs. For example, you can set the `Max errors` value to define the maximum number of allowable errors before stopping the test (0=continue to run on infinite errors).
- **Advanced**—provides overrides and locks that control which option settings (system-level, group-level, or device-level) take precedence.
- **Option Files**—creates or selects an option file which can be loaded, stored, or removed.

---

**Note** – These dialog boxes are described in detail in Appendix A.

---

---

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

---

## 7. Modify group-level and device-level options (if desired).

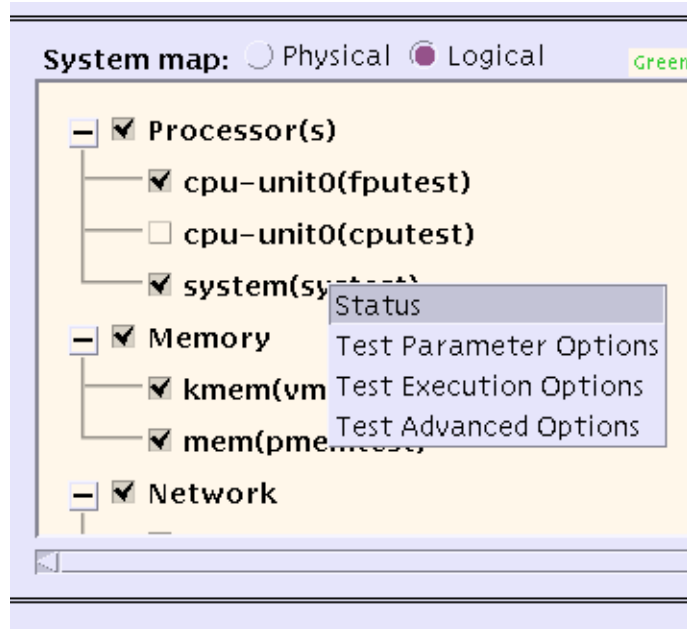


FIGURE 4-9 Test-Specific Options

*About group-level and device-level options:*

Group-level and device-level options are testing attributes that accompany each test and pertain to a specific device (or device group) only.

*How to do it:*

Access the group-level and device-level menus by placing your pointer over a device (or device-group) and right-clicking on it.

A menu appears offering access to the following dialog boxes (FIGURE 4-9):

- Status—displays a list showing each test instance and a test pass and fail value for each instance.
- Test Parameter Options—Shows device configuration information and provides the controls for defining specific test options. This menu is different for each kind of test. For example, the Test Parameter Options for the memory test (`pmemtest`) allow you to specify the amount of memory to test, while the network test (`nettest`) provides a setting to define a target host for test packets. Refer to the *SunVTS Test Reference Manual* for descriptions of all the Test Parameter Options. Once you change a Test Parameter Option, apply the change to the desired level:
  - Within Instance—applies the option settings to this test instance (test instance 1 for example) with:
    - Apply—to this device only.

- Apply to Group—to all the same instances, of this device type, in the group.
- Apply to All—to all the same instances of the same device type on the system.
- Across All Instances—applies the option settings to all test instances (for example, instance 1 and 2) with:
  - Apply—for all test instances for this device.
  - Apply to Group—for all test instances for the same device type in the group.
  - Apply to All—for all test instances for the same device type on the system.
 See “Test Parameter” on page A-19 for more information.
- Test Execution Options—allows you to define *how* a test runs. For example, you can set the `Max errors` value to define the maximum number of allowable errors before stopping the test (0=continue to run on infinite errors). The options in this menu are the same for each kind of test.
- Test Advanced Options—Provides a setting to “lock” the option settings for an individual device so that the system level setting will not affect the option setting for this test. See “Preserving Test Options with Locks and Overrides” on page 4-23 for more details.
- Schedule Options (group level only)—sets the maximum number of tests you want to run at the same time in the same group.

---

**Note** – For descriptions of the options presented in these dialog boxes see, “SunVTS Dialog Boxes” on page A-8.

---

## ▼ To Run, Monitor, and Stop a Test Session

### 1. Start the test session by clicking on the Start Button.

Depending on how you configured the options in the Test Execution menus, testing continues until:

- A test error is detected (this is the default behavior). If you enter a value for Run On Error, SunVTS continues to run until the specified number of errors is reached.
- Testing has reached the Max Passes value. The default is to run indefinite test passes. When individual test passes are complete, SunVTS continues to display the status testing for 5 seconds. If you do not select additional devices to test in the 5 second interval, SunVTS enters an idle state.
- Testing has reached the Max Time value. The default is to run indefinitely.
- You select the Stop button.

## 2. Monitor the overall test session status by watching the Status panel.

The Status panel indicates that a test session is running by displaying *testing* for the Testing status. The system passes, cumulative errors, and elapsed test time will increment.

## 3. Monitor specific tests by watching the System Map.

Watch for:

- An asterisk—displayed next to each device indicating the device that is currently under test. Depending on how you configured SunVTS (with the System Concurrency option), single or multiple devices are tested concurrently.
- Colors – Indicate the state of devices as follows:
  - Black – indicates that the device is not tested, or the first pass is not complete.
  - Green – indicates that at least one test pass completed with no errors detected.
  - Red – indicates that at least one error was detected.

## 4. Check for test messages by watching the Test Message panel.

The Test Message panel displays test messages under the following circumstances:

- When a test error occurs.
- If you enable trace mode (from the Commands→Trace test menu bar selection) and set the trace to display in the window, all of the test system calls are displayed.
- If you enable verbose mode (through the Test Execution dialog box), verbose messages are displayed.

## 5. Stop the test session (if desired) by clicking the Stop button.

# ▼ To Review and Reset Test Session Results

## 1. Review the logs.

*About logs:*

SunVTS provides access to four log files:

- SunVTS test error log – contains time-stamped SunVTS test error messages. The log file pathname is `/var/opt/SUNWvts/logs/sunvts.err`. This file is not created until a SunVTS test failure occurs.
- SunVTS kernel error log – contains time-stamped SunVTS kernel and SunVTS probe errors. SunVTS kernel errors are errors that relate to running SunVTS, and not to testing of devices. The log file pathname is `/var/opt/SUNWvts/logs/vtsk.err`. This file is not created until SunVTS reports a SunVTS kernel error.



- SunVTS information log—contains informative messages that are generated when you start and stop SunVTS test sessions. The log file pathname is `/var/opt/SUNWvts/logs/sunvts.info`. This file is not created until a SunVTS test session runs.
- Solaris system message log—is a log of all the general Solaris events logged by `syslogd`. The pathname of this log file is `/var/adm/messages`.

*How to do it:*

**a. Click the Log button.**

The Log file window is displayed.

**b. Specify the log file that you want to view by selecting it from the top button in the Log file window.**

The content of the selected log file is displayed in the window.

**c. With the three lower buttons you can:**

- Print the log file—a dialog box appears for you to specify your printer options and printer name.
- Delete the log file—the file remains displayed, but will be gone the next time you try to display it.
- Close the Log file window—the window is dismissed.

---

**Note** – Take the length of the log file into consideration before you print it.

---

## **2. Interpret SunVTS messages.**

There are many messages for each of the many tests that run under SunVTS. For this reason, it is difficult to document the implication of each message that you might see. Most messages contain all the explanatory text that is possible for each event that is encountered. Some messages are information messages (INFO, VERBOSE,

WARNING) which do not represent an error, and some messages are error messages (ERROR and FATAL), indicating that the test detected a failure. This section explains generic test message information.

SunVTS information message example:

```
04/24/00 17:19:47 systemA SunVTS5.1: VTSID 34 disktest.VERBOSE
c0t0d0: "number of blocks 3629760"
```

SunVTS error message example:

```
05/02/00 10:49:43 systemA SunVTS5.1: VTSID 8040 disktest.
FATAL diskette: "Failed get_volmgr_name()"
Probable_Cause(s):
(1)No floppy disk in drive
Recommended_Action(s):
(1)Check the floppy drive
```

Whenever possible, the error message supplies a message type (as shown in the following table), followed by the message text, one or more probable causes, and one or more recommended corrective actions.

The following table describes the types of messages you might encounter. All messages are displayed in the Message panel, and most are logged in one of the SunVTS logs (Info or Error).

**TABLE 4-1** Message Types

Message Type	Log file	Description
INFO	Info log	Displayed when a nonerror test event occurs.
ERROR	Error and Info logs	Displayed when a test detects an error, often an indication of a problem with a specific function or feature of the device under test.
FATAL	Error and Info logs	Displayed when a test detects a severe error that might cause the test to exit; such as a failure to open a device. These errors are indicative of hardware failures.
VERBOSE	Not logged	Test progression messages that are displayed when the verbose feature is enabled.
WARNING	Info log	Displayed when something affects the test that is not an error; such as a device that is busy.

You can write scripts that watch for certain SunVTS events as they are logged in the log files, and which trigger an action when a particular message string occurs. If you

plan to do this, refer to the following table of SunVTS message syntax and descriptions.

SunVTS Message Syntax:

```
timestamp hostname "SunVTS<version_id>:" [VTSID vts_msgid
modulename [ .submodulename] [ .instnum] [ (#P) ] .vts_msgtype]
[ device_pathname: ] msg_text
```

TABLE 4-2 SunVTS Message Syntax Descriptions

Message Item	Descriptions
<i>timestamp</i>	Represents the date in <i>mm/dd/yy</i> format, and the time in <i>hh:mm:ss</i> format, when the message was logged.
<i>hostname</i>	Name of the system under test.
<i>SunVTSversion_id</i>	Displays the string SunVTS followed by the version of SunVTS, for example SunVTS5.1.
VTSID <i>vts_msgid</i>	Displays the string VTSID followed by the ID of the message.
<i>modulename</i>	Name of the module (usually the test name or <code>vts:k</code> ) that generated the message.
<i>submodulename</i>	Name of the subtest, if applicable.
<i>instnum</i>	Number of the test instance.
( <i>#P</i> )	Processor ID for which the test is bound (applicable if a test is assigned to a specific processor with the processor affinity option).
<i>vts_msgtype</i>	Identifies the message with one of the following message types: ERROR, FATAL, INFO, VERBOSE, or WARNING.
<i>device_pathname</i>	Name of the device under test.
<i>msg_text</i>	The text of the message. This often includes the error, a probable cause, and a recommended action.

**Note** – As of SunVTS 5.0, the `VTS_OLD_MSG` variable is no longer supported. It was used to display test messages in a pre-SunVTS 4.0 format, usually because a script relies on the older format. Now your scripts must accept the current SunVTS message format.

### 3. Reset the test session results (if desired).

When you are done reviewing the information in the main window, you can clear the current results by clicking on the Reset button. This resets the information in the Status panel. This does not reset your test options.

---

## 4.2 Using Additional Features With the SunVTS CDE UI

This section describes additional SunVTS features using the SunVTS CDE user interface:

- “To Use the Automatic Configuration Feature” on page 4-19
- “Connecting to Another Host” on page 4-20
- “Using the Email Notification Feature” on page 4-21
- “Controlling the Size of the Log Files” on page 4-21
- “Tracing a Test” on page 4-22
- “Preserving Test Options with Locks and Overrides” on page 4-23
- “Scaling Your Test Session” on page 4-24
- “Using the Debugging Features” on page 4-25
- “Saving a Test Session Configuration for Repeated Use (Option Files)” on page 4-26
- “Using the Auto-Start Feature” on page 4-28
- “Suspending and Resuming a Test Session” on page 4-29
- “Recording and Replaying a Test Session” on page 4-29

### 4.2.1 Using the Automatic Configuration Feature

The Automatic Configuration feature simplifies the SunVTS configuration steps by automatically assigning a predetermined set of test options. This feature provides the following benefits:

- Consistent testing—as you test the system from time to time, you know that the same test options are applied. You also get diagnostic consistency when you test multiple systems of the same platform type.
- Reliable testing—provides assurance that any errors that you might encounter are not due to incorrectly set test options.
- Simplified configuration—there are many options for the numerous SunVTS tests. The Automatic Configuration feature takes the guesswork out of assigning these options.
- Time Saving—You can select the Automatic Configuration feature without manually setting the options and start testing right away.

## ▼ To Use the Automatic Configuration Feature

**1. Prepare any “intervention” devices and select the Intervention checkbox.**

See “Preparing Devices for Testing” on page 3-4.

**2. Select Auto Config from the Select Test Mode panel.**

The Automatic Configuration dialog box is displayed (FIGURE 4-10).

**3. Select one of the following choices from the Automatic Configuration dialog box:**

- **Comprehensive**—sets all the test options for thorough testing. All applicable functional tests are enabled. The test session validates the full functionality of the system and provides you with assurance that the system is hardware fault free.
- **Confidence**—sets all the test options such that there is limited functional test coverage as compared to the Comprehensive level of coverage. The Confidence level testing verifies the main functionality of the system in a shorter amount of time.

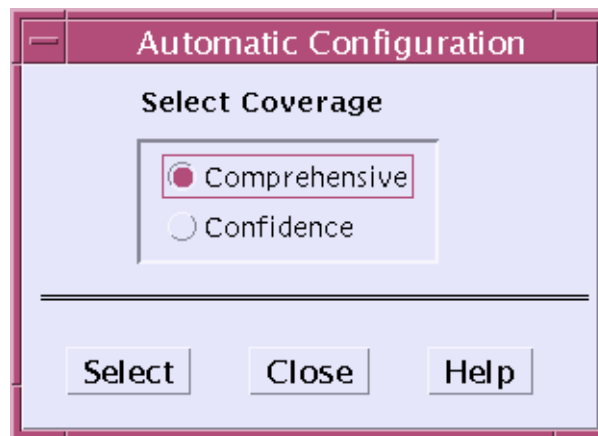


FIGURE 4-10 Automatic Configuration Dialog Box

---

**Note** – The Help button displays a dialog box with Auto Configuration information and instructions.

---

**4. Click on Select in the dialog box.**

SunVTS applies the predefined options and is ready to test selected devices.

**5. (Optional) Select or deselect the devices that you want to test in the System Map.**

This step is optional. For full system testing, use the devices that are selected by the Automatic Configuration feature.

**6. Click the Start button in the SunVTS main window.**

The test session runs with the predetermined set of options based on the level of coverage (comprehensive or confidence) that you selected. The AC Coverage (Confidence or Comprehensive) is displayed in the Status panel.

## 4.2.2 Connecting to Another Host

You can connect the SunVTS user interface on your local system to a SunVTS kernel that is running on another system in your network. Once you are connected to the remote system, all of the controls in the CDE UI control the testing of the remote system.

### ▼ To Connect to Another Host

**1. Make sure that the SunVTS kernel is running on the remote system.**

You can do this in several ways. You can remote login (using `rlogin` or `telnet`) to the remote system and run the `vtsk` command.

---

**Note** – On the remote system, you must login (or switch user) as a user that has SunVTS privileges (superuser by default) before you can start the SunVTS kernel (`vtsk`).

---

---

**Note** – You can only connect to a system running the same version of SunVTS.

---

**2. On the local system (the system running the SunVTS UI), click on the Connect to Host button.**

The Connect to Host dialog box is displayed.

**3. Specify the name of the remote system to which you want to connect.**

Enter the hostname in the Connect to Host field. As you connect to various remote systems, SunVTS keeps track of those systems and displays them in the Hostname List. You can reconnect to one of those hosts by double-clicking on the host name in this list. This list is only maintained during your current SunVTS session. When you quit SunVTS, the Hostname List is cleared.

Any number of UIs can connect to a SunVTS kernel and each UI displays a synchronous view of SunVTS.

**4. Click Apply in the Connect to Host dialog box.**

The SunVTS UI now controls the SunVTS testing of the remote system. The remote system hostname is displayed in the Status panel, confirming the connection.

## 4.2.3 Using the Email Notification Feature

You can have SunVTS send the test status messages to you using email.

### ▼ To Enable Email Notification

**1. Select Notify from the Options menu in the Menu bar.**

The Notify Options dialog box is displayed.

**2. Select one of the following from the Send Email pull-down menu:**

- disabled—disables the email notification feature.
- now—send email immediately one time, then disable the notification feature.
- on error—send email when an error occurs.
- periodically—send email at regular intervals as specified in the Log Period.
- on\_error&periodically—send email periodically and when an error occurs.

**3. Enter an email address in the email address field.**

**4. If you selected Periodically or on\_error&periodically, specify the log period in minutes (1-99999).**

**5. Click on Apply.**

You will receive the SunVTS information messages in email based on your notification criterion.

## 4.2.4 Controlling the Size of the Log Files

The log file sizes of the SunVTS test error log (`/var/opt/SUNWvts/logs/sunvts.err`), the SunVTS kernel error log (`/var/opt/SUNWvts/logs/vtsk.err`) and the information log (`/var/opt/SUNWvts/logs/sunvts.info`) are each limited to a maximum size of 1 Mbyte by default. When the log file reaches the maximum size, the content is moved to a file called *logfilename.backup*. Additional events are added to the main log files. If the log file reaches the maximum size again, the content is moved to the backup file, overwriting the earlier backup file content. Only one backup file is maintained for each log file.

You can modify the log file maximum size specification.

### ▼ To Modify the Log File Size Limits

**1. Select Option→Thresholds from the Menu bar.**

The Threshold Option dialog box is displayed.

**2. Use the arrows to increase the Max System Log Size number.**

Select a number from 1 to 5 (Mbytes). For example, if you choose 5 Mbytes, SunVTS will use up to 30 Mbytes for log files (there are three log files, each may also have an equal sized backup file).

**3. Click on Apply.**

## 4.2.5 Tracing a Test

You can use the Trace test option to create a log of every system call made when a test is running. This feature logs the system calls using the standard UNIX command `truss`. The trace messages logged by this feature give you a powerful debugging tool for isolating the specific cause of an error.

### ▼ To Enable Tracing

**1. Select Commands→Trace test from the Menu bar.**

The Trace Test window is displayed.

**2. Choose a test to trace from the list of test names and then select Apply, or double-click a test name to enable tracing and close the window.**

When you select a test, system call tracing is enabled immediately. If the test is already running when you select it, tracing begins immediately and the trace test messages appear in the SunVTS console window.

---

**Note** – You can only select one test at a time to trace when using the SunVTS CDE user interface.

---

---

**Note** – When using the Physical map view, you can only see one level in the hierarchy (such as one controller on a system board). Therefore, some devices that are several levels down are not available to trace. Use the Logical view to select tracing for these devices.

---

**3. Select File and click Apply to write the trace test messages to a file called `/var/opt/SUNWvts/logs/sunvts.trace`.**

### ▼ To Disable Tracing

**1. Bring up the Trace Test window again, click on the highlighted test name, and click Apply.**

This action deselects the test and disables tracing.



## 4.2.6 Preserving Test Options with Locks and Overrides

With locks and overrides you can preserve or override the options you set at the system level, group-level, and device-level (see “Modify the system-level test options (if desired).” on page 4-10).

### 4.2.6.1 Locks

Normally, when you change an option setting at the system level or group-level, the new option setting propagates down to all lower levels. Enabling the lock prevents an option setting made at a higher level from affecting the lower-level option setting.

---

**Note** – The override option nullifies lower-level locks.

---

#### ▼ To Set (or Unset) a Lock

1. **Open the Test Advanced Options dialog box from the group-level or device-level you want to lock.**

To access this dialog box, place your pointer over the group or device and right-click. Then select Test Advanced Options from the menu.

2. **Click the Enable button to set the lock (or disable button to unset the lock).**

With the lock set, you can make option setting changes at higher levels without affecting the option settings of the locked device.

### 4.2.6.2 Overrides

You can use overrides to void the lock protection. Setting the system-level override nullifies *all* locks, and setting a group-level override nullifies all the locks below that group.

#### ▼ To Set (or Unset) an Override

1. **Open the Advanced Option dialog box at one of the following levels:**
  - System level—select Option→Advanced from the Menu bar.
  - Group level—place your pointer over the group and right-click. Then select Test Advanced Options.
2. **Enable the Override setting (or disable the Override setting to unset it).**
3. **Click Apply.**

## 4.2.7 Scaling Your Test Session

You can use a combination of options to scale (increase or decrease testing intensity) to meet your diagnostic needs. For example, you can modify test options so that each test instance runs simultaneously, thus increasing the stress level for a single or multiprocessor system. The following procedures can be used individually or in combination to scale the test session.

### ▼ To Modify the Number of Test Instances

You can scale your test session by running multiple copies of the same test on your devices. Each copy is called a *test instance*. Each test instance is a separate process of the same test. The number of test instances can be set at the system level, group level, and the device level as follows:

- 1. Open the Test Execution dialog box at one of the following levels:**
  - System level—select Option→Test Execution from the Menu bar.
  - Group level—place your pointer over the group and right-click. Then select Test Execution Options.
  - Device level—place your pointer over the device and right-click. Then select Test Execution Options.
- 2. Use the arrow buttons to increase or decrease the number of instances.**
- 3. Click Apply.**

### ▼ To Modify Test Concurrency Options

The test concurrency option sets the number of tests that run at a given time during a test session. You can set this value to one to run one test at a time, or increase the test concurrency value thus increasing the stress level of your test session.

There are two options in the Schedule Option dialog boxes (at the system level and group level) to scale your test session:

- System Concurrency—sets the maximum number of tests that you can run at the same time at the system level. It overrides the Group Concurrency option.
  - Group Concurrency—sets the maximum number of tests to run simultaneously in the same group. This option is available at the system level and the group level.
- 1. Open the Schedule dialog box from one of the following levels:**
    - System level—select Option→Schedule from the Menu bar.
    - Group-level—place your pointer over the group and right-click. Then select Schedule Options.
  - 2. Use the arrow buttons to increase or decrease the System Concurrency and Group Concurrency values.**

### 3. Click Apply.

## ▼ To Bind a Test to a Processor with the Processor Affinity Option (for Multiprocessor Systems)

By default, the Solaris kernel assigns each test instance to whichever processor is available at the time. On multiprocessor systems, you can bind (assign) a test instance to a particular processor as follows:

### 1. Open the Test Execution dialog box at one of the following levels:

- System level—select Option→Test Execution from the Menu bar.
- Group-level—place your pointer over the group and right-click. Then select Test Execution Options.
- Device-level—place your pointer over the device and right-click. Then select Test Execution Options.

---

**Note** – The Processor Affinity field is only displayed on multiprocessor systems.

---

---

**Note** – The Processor Affinity option is not available for processor tests such as `cputest` and `fputest`. These tests are associated with each individual processor on your system and you cannot bind these tests to a different processor.

---

### 2. Select a Processor from the Processor Affinity field.

By default, the test instances are not bound to any processor. You can either keep this unbound or select a specific processor to bind or select option for “sequential” binding.

Selecting the “sequential” option binds the test instance(s) to different processors in the system as the test instances are created. Even if there is a higher number of processors than instances, the sequential binding will eventually bind the test instances to all the processors in subsequent test passes.

### 3. Click Apply.

## 4.2.8 Using the Debugging Features

You can enable certain options that cause the SunVTS test session to output more test data than when you run the session with the default option values. The following procedures describe how to enable these features.

## ▼ To Enable (or Disable) Verbose and Core File Options

1. **Open the Test Execution dialog box at one of the following levels:**
  - System level—select Option→Test Execution from the Menu bar.
  - Group-level—place your pointer over the group and right-click. Then select Test Execution Options.
  - Device-level—place your pointer over the device and right-click. Then select Test Execution Options.
2. **Enable (or disable) one or more of the following options:**
  - Verbose—displays verbose messages indicating when the test is starting and stopping during the test session.
  - Core File—when enabled, creates a core file when a test session results in a core dump. The core file is called `sunvts_install_dir/bin/core.testname.xxxxxx`, where *testname* is the test that dumped the core, and *xxxxxx* is a character string generated by the system in order to make the file name unique.

---

**Note** – The Trace feature also provides additional test data. See “Tracing a Test” on page 4-22.

---

## 4.2.9 Saving a Test Session Configuration for Repeated Use (Option Files)

You can use the Option Files feature to save the current set of selected devices and test options for reuse. This is a convenient feature when you plan to perform the same test session configuration repeatedly.

The configuration information is saved in a filename of your choice in the `/var/opt/SUNWvts/options` directory.

The option file could be saved in host-dependent format or host-independent generic option format. If the option file is stored in host-dependent format, it can be used on other hosts only if the device configurations match. Alternately, the generic option format is designed to work on multiple hosts with different device configurations. The generic option format may save one set of options for one test which can then be applied to all devices (irrespective of the number of devices) for that test in that machine or other machines. Note that this feature is useful when the number of devices of a specific test is machine-dependent.

While saving the configuration in host-independent generic option format, select at most, one device per device type. If multiple devices of similar type have been selected, any one device will be chosen automatically for being saved. When the same option file is loaded, all devices of that device type will be selected and configured with the same test options.

The host-independent option file could be saved for either normal load or cumulative load. The cumulative generic option file can be used to do incremental configuration by loading one after another. Loading a cumulative option file does not deselect already selected test devices. Alternately, loading of a non-cumulative option file first deselects all tests, then only selects the devices that are defined in the option file.

The host-independent generic option file save feature also has an option to save the total number of instances. If this option is chosen, the same number of instances will be forced during load time.

The option file contains information about system level options and test level options. Option file creation is primarily intended to be done from a GUI. However, you can edit option files with any text editor to accommodate specific needs, if needed. See Appendix C for information on editing generic option files.

## ▼ To Create an Option File

### **1. Configure SunVTS for the test session that you want to save.**

See “To Configure SunVTS for a Test Session” on page 4-4.

### **2. Select Options→Option files from the Menu bar.**

The Option Files dialog box is displayed.

### **3. Select an option file name in the Option File field.**

### **4. Select Host Dependent or Generic.**

For generic options, select Enable/Disable Cumulative option, and select Enable/Disable Save Number of Instances.

### **5. Select Store.**

### **6. Repeat Step 1 through Step 5 for each test session configuration you want to save.**

### **7. Select Close in the Option Files dialog box.**

The Option Files dialog box is closed, and your test session configurations are saved for future use.

## ▼ To Load an Option File

---

**Note** – Loading an option file that was created on another system is permitted, but you must make sure that the configuration is valid for the system you plan to test.

---

---

**Note** – Do not load an option file that was created in a 64-bit environment into a 32-bit environment.

---

1. **Select Options→Option files from the Menu bar.**

The Options Files dialog box is displayed.

2. **Select an option file from the Option file list.**

3. **Click Load.**

The test session configuration is loaded into SunVTS and the name of the option file is displayed in the Status panel. You can use this configuration, or modify it before you start the test session.

## ▼ To Remove an Option File

1. **Select Options→Option files from the Menu bar.**

The Options Files dialog box is displayed.

2. **Select the option file that you want to delete from the Option File List.**

3. **Click Remove.**

4. **Click Close.**

## 4.2.10 Using the Auto-Start Feature

You can use the auto-start feature, along with saved option files, to simplify the process of configuring a test session and running it. When the auto-start feature is turned on in an option file, SunVTS automatically starts the test session when the option file is invoked, eliminating the need to select the start button.

## ▼ To Configure SunVTS to use the Auto-Start Feature

1. **Select Options→Schedule from the Menu bar.**

The Schedule dialog box is displayed.

2. **Enable the Auto Start setting in the Schedule dialog box.**

3. **Create an option file as described in “To Create an Option File” on page 4-27.**  
The option file that you create is configured to use the auto-start feature.

## ▼ To Use the Auto-Start Feature

1. **Select Commands→Quit SunVTS→UI and Kernel.**

You must quit SunVTS and restart it for the auto-start feature to work.

2. **Use the following command to restart SunVTS from the command-line, specifying the option file that you created:**

```
# /opt/SUNWvts/bin/sunvts -o option_file
```

SunVTS displays the main window and automatically starts the test session.

## 4.2.11 Suspending and Resuming a Test Session

You can suspend a test session while it is running. For example, you might want to look at messages on the Message panel that have scrolled out of view, or you may want to view and print a log file.

### ▼ To Suspend and Resume a Test Session

1. **Select Commands→Suspend from the Menu bar.**

“Suspended” is displayed in the Status panel, and the test session pauses until you resume it.

2. **Select Commands→Resume from the Menu bar.**

“Testing” is displayed in the Status panel, and the SunVTS kernel resumes the suspended test session.

## 4.2.12 Recording and Replaying a Test Session

You can use the Record and Replay feature to record a SunVTS test session. Only one test session recording is saved at a time.

The events are recorded in a file called `/var/opt/SUNWvts/vts_replay_file`.

Once a test session is recorded, you can use the recorded events to drive the SunVTS kernel so it reproduces the recorded sequence of events at a later time.

---

**Note** – The Record and Replay feature closely reproduces the sequence of events, but it cannot reproduce the time periods of these events because the execution times vary from one run to another.

---

---

**Note** – Do not edit the `/var/opt/SUNWvts/vts_replay_file`.

---

## ▼ To Record and Replay a Test Session

### 1. Configure SunVTS for the test session that you want to run.

See “To Configure SunVTS for a Test Session” on page 4-4.

### 2. Select Commands→Start testing with record from the Menu bar.

The test session runs, and the events are recorded. When testing stops, the recorded session is available to replay.

### 3. Select Commands→Replay to replay the test session.

The kernel reruns the same tests, with the same configuration that you specified in Step 1.

---

**Note** – During the replay, the kernel is actually rerunning the tests, not just re-displaying the recorded session.

---

---

## 4.3 Scheduling Test Sequences With the Deterministic Scheduler

SunVTS provides a Deterministic Scheduler that you can use to define what tests run and in what order. If you do not use this feature, selected tests run in an order that is dynamically determined by SunVTS kernel.

### 4.3.1 Deterministic Scheduler Overview

The Deterministic Scheduler is available as a dialog box from the SunVTS CDE interface menu-bar area.



The following list provides an overview of the steps you perform to start, run, and quit the Deterministic Scheduler feature. Detailed instructions follow.

1. Start SunVTS with the SunVTS CDE user interface.
2. Start the Deterministic Scheduler.
3. Create one or more *tasks*. A task is a SunVTS test session configuration (one or more selected SunVTS tests, test options, and specified test mode).
4. Create the *sequence*. A sequence is a group of tasks specified to run in a certain order.
5. Define the number of times (*loops*) that sequence runs.
6. Start the sequence.
7. When the sequence is done, reset the Deterministic Scheduler if desired.
8. Quit the Deterministic Scheduler feature.

## 4.3.2 Getting Started With the Deterministic Scheduler

1. **Start SunVTS with the SunVTS CDE user interface as described in “Starting SunVTS” on page 3-1.**
2. **Start the Deterministic Scheduler by selecting DSched→start DS from the SunVTS menu bar.**

The Deterministic Scheduler Dialog box is displayed (FIGURE 4-11).

---

**Note** – When you start the Deterministic Scheduler, many of the commands from the main SunVTS window such as Start, Stop, Resume, Suspend, Record, Replay, Load, and Set are not permitted. When you stop the Deterministic Scheduler, these SunVTS features become functional.

---

---

**Note** – The Help button displays a dialog box with Deterministic Scheduler information and instructions.

---

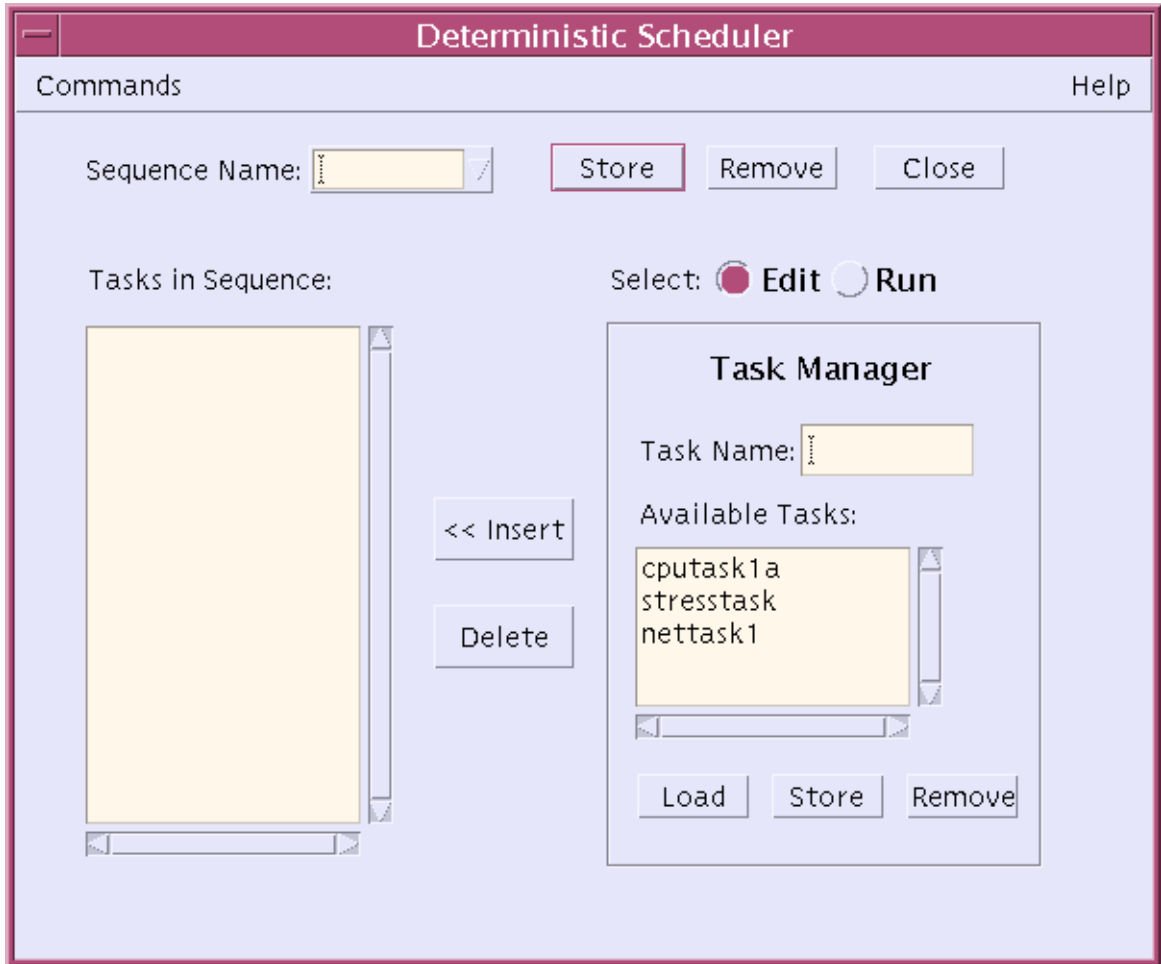


FIGURE 4-11 Deterministic Scheduler Dialog Box

**3. Define your task(s) as described below:**

**a. In the Deterministic Scheduler dialog box, click on the Edit button.**

The Deterministic Scheduler displays the Task Manager panel and shows a list of existing tasks (if any have been created) in the Available Tasks list. You can use existing tasks or create new tasks. If this is the first time that the Deterministic Scheduler is used, you must create your own.

- b. In the SunVTS main window, configure a test session (select one or more devices and your desired test mode).**

If you assign more than one test to a task, the order of the tests within this task will be determined by SunVTS. If you want to control the order of every test, assign only one test for each task.

When the Deterministic Scheduler is started, it changes the default values of Max System Passes from 0 (run *indefinitely*) to 1 (run *one* pass). This is necessary so that each task only runs selected tests once and proceeds to run subsequent tasks in the sequence.

- c. In the Deterministic Scheduler dialog box, enter a task name.**

- d. Click on Store in the Task Manager panel.**

Your task appears in the Available Task list.



**FIGURE 4-12** Tasks Defined in the Task Manager Panel

- e. Repeat Step a through Step d to create additional tasks.**

- f. To see the tests configured for a particular task, select the task from the Available task list and press Load.**

The SunVTS main window shows the SunVTS test configuration that is associated with the selected task.

- g. If you want to modify a task: select it; load it; change the test configuration in the SunVTS main window; and press Store in the Task Manager panel.**
- h. If you want to delete a task, select it and press Remove in the Task Manager panel.**

The task is removed from the Available Task list.

**4. Define the sequence as follows:**

- a. Select a task from the task list.**

- b. Press the Insert button.**

The task is added to the sequence list.

- c. Repeat the previous steps until all desired tasks are added to the sequence list in the order that you want them to run.**

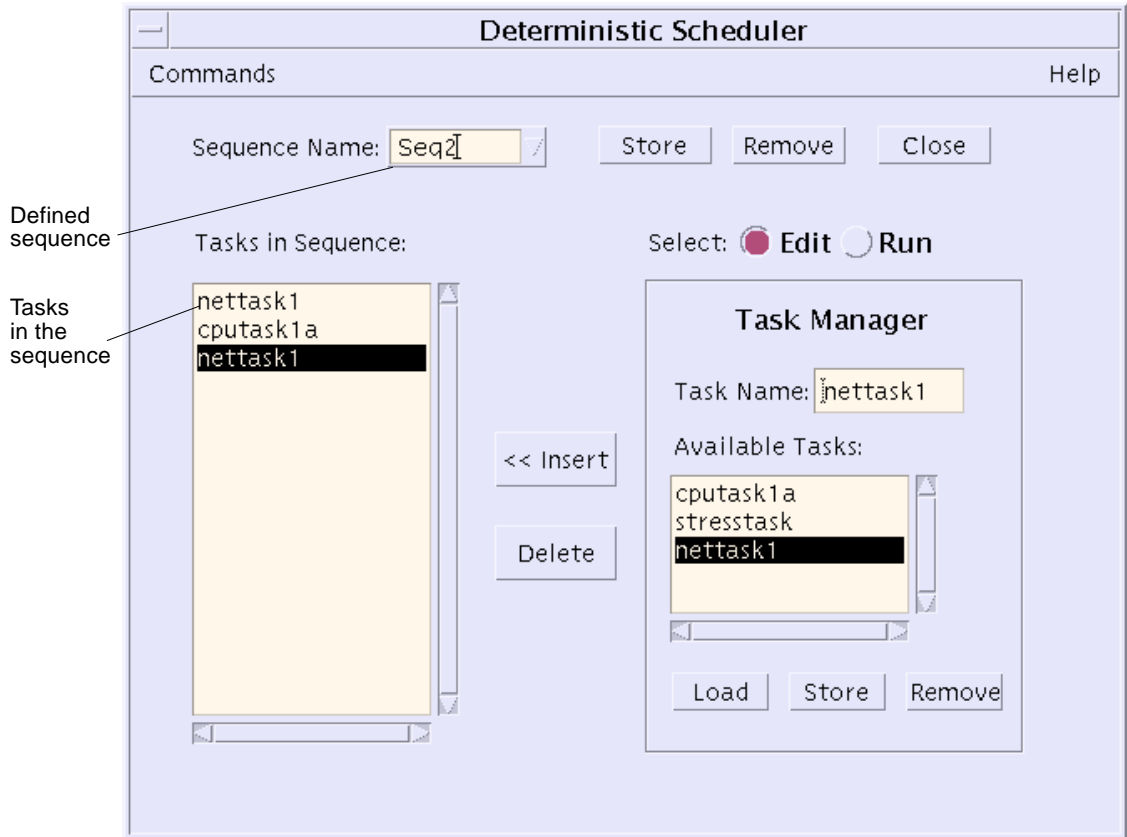
You can add the same task more than once.

When you begin testing, the tasks run from the top of the list down.

- d. If you want to remove a task from the sequence list, select it and press Delete.**

- e. To save the sequence for future use, enter a name in the Sequence name field and press Store in the Task Manager panel.**

The sequence will be saved for easy retrieval for future testing.



**FIGURE 4-13** Deterministic Scheduler Dialog Box

**f. With the Sequence selected, press the Run button.**

The right-hand panel displays the Sequence Runner panel (FIGURE 4-14), providing access for you to define the number of times (loops) for the sequence to run and to start running the sequence.

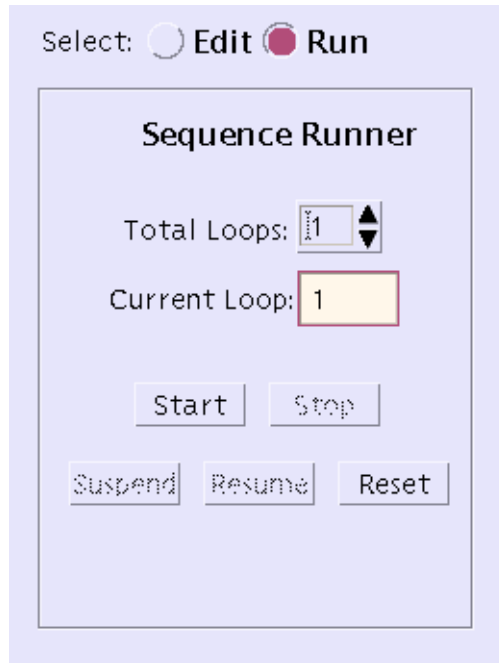


FIGURE 4-14 Sequence Runner Panel

**5. Define the sequence loops.**

- a. **Increment or decrement the Total Loop value to the desired number of times you want the sequence to run.**

**6. Press the Start button to start the sequence.**

While testing is underway you can:

- monitor the test progress in the main SunVTS window. All test results are logged in the SunVTS log files.
- close the Deterministic Scheduler dialog box by pressing the close button or by selecting **Commands**→**Quit Options**→**Quit UI only** from the Deterministic Scheduler dialog box. Testing will continue, and results will be displayed in the SunVTS main window. You can reopen the Deterministic Scheduler dialog box from the SunVTS main window.
- suspend and resume the sequence by pressing the suspend and resume buttons in the Deterministic Scheduler dialog box.

**7. Use the Reset button to reset the SunVTS main window and the Deterministic Scheduler.**

The SunVTS main window test statistics and the Deterministic Scheduler current loop count are reset.

**8. Quit the Deterministic Scheduler when you are done by selecting  
Commands→Quit Options→Quit DS and UI.**

The Deterministic Scheduler and the Deterministic Scheduler dialog box are dismissed.





# Using the SunVTS TTY User Interface

---

This chapter describes how to run a test session using the SunVTS TTY UI. The procedures are written in a step-by-step form so you can use them to become familiar with SunVTS using the SunVTS TTY UI. This chapter is divided into the following sections:

- “Getting Started With the SunVTS TTY UI” on page 5-1
- “Using Additional Features With the SunVTS TTY UI” on page 5-17
- “Scheduling Test Sequences With the Deterministic Scheduler” on page 5-27

---

**Note** – This chapter assumes that you have already performed the following procedures:

- Installed SunVTS as described in “Installing and Removing SunVTS” on page 2-1
  - Started SunVTS as described in “Starting SunVTS” on page 3-1
  - Prepared your system for testing as described in “Preparing Devices for Testing” on page 3-4
- 

See Appendix A for descriptions of each SunVTS window and dialog box.

---

## 5.1 Getting Started With the SunVTS TTY UI

This section describes how to use the fundamental features of the SunVTS TTY UI to perform diagnostic testing on a system. For information about using more advanced features, see “Using Additional Features With the SunVTS TTY UI” on page 5-17.

## 5.1.1 The SunVTS TTY UI Main Window and Keyboard Commands

The SunVTS TTY UI is an ASCII based, screen-oriented interface that does not require a graphical window environment. Most of the control and monitoring options that are available in the SunVTS graphical interfaces are also available in the TTY interface.

You can use the TTY interface to run SunVTS from a serial terminal, or through a graphical window tool such as a shell tool, command tool, or CDE terminal window.

The procedures in this section explain how to use the TTY main window:

- “To Configure SunVTS for a Test Session” on page 5-4
- “To Run, Monitor, and Stop a Test Session” on page 5-13
- “To Review and Reset Test Session Results” on page 5-14



**FIGURE 5-1** SunVTS TTY Main Window

**TABLE 5-1** SunVTS TTY Main Window

Window Item	Description
Control panel	Provides access to SunVTS controls.
Test_Groups panel	shows the testable devices (categorized in groups) and provides access to their test options.
Status panel	Displays SunVTS test status information.
Message console panel	The area where test messages are displayed. No user input is accepted in this panel. You can use up and down arrow keys to scroll through the messages in this panel.

---

## 5.2 Navigating the SunVTS TTY UI

Use keyboard commands instead of mouse and pointer action to control SunVTS in the TTY UI. The following table lists the keyboard commands and explains how to use them.

**TABLE 5-2** TTY Keyboard Commands

Key	Description of Action
Tab	Moves you from one panel to another. For example, if you are working in the Control panel and press the Tab key, the focus (or highlight) shifts to the Status panel. Highlighted panels are bordered by asterisks (*).
Arrow keys	Moves between selections within a panel
Return	<ul style="list-style-type: none"><li>• Displays a menu</li><li>• Selects and applies a choice or a command in a menu</li></ul>
Spacebar	Selects or deselects a checkbox: <ul style="list-style-type: none"><li>• [ * ]= selected</li><li>• [ ]= deselected</li></ul>
Backspace key	Deletes text in a text field
Escape	Discards a pop-up menu or window
Control-F	Scrolls forward in a scrollable window
Control-B	Scrolls backward in a scrollable window
Control-X	Quits the TTY user interface but leaves the SunVTS kernel running
Control-L	Refreshes the TTY window

### ▼ To Configure SunVTS for a Test Session

1. **Identify the current state of SunVTS in the Status panel.**

```

*****Status*****
                System_status: idle
System_passes:0      Total_errors:0
                Elapsed_time:000:00:00
                Status_view
                Passes Errors

Processor(s)
  cpu-unit0(fputest)      0      0
  cpu-unit0(fputest)      0      0
Memory
  kmem(vmemtest)         0      0
  kmem(vmemtest)         0      0
  mem(pmemtest)          0      0
  mem(pmemtest)          0      0
Network
  hme0(nettest)          0      0
  hme0(nettest)          0      0
Graphics
  m640(m64test)          0      0
IDE-Devices(uata0)
  c0t0d0(disktest)       0      0
  c0t0d0(disktest)       0      0
OtherDevices
  ecpp0(ecpptest)        0      0
*****

```

**FIGURE 5-2** Status Panel (TTY)

- **System\_status**—provides the following status of SunVTS:
  - **Idle**—no testing is taking place
  - **Testing**—a test session is running
  - **Suspend**—a test session is suspended
  - **Replay**—a previously recorded test session is displayed
  - **Stopping**—is momentarily displayed when a test session stops
- **System\_passes**—displays the total number of successful system passes (a system pass is when all tests have run once).
- **Total\_errors**—the total number of errors from all tests.
- **Elapsed\_time**—indicates the total number of hours, minutes, and seconds that elapsed during testing.
- **List of devices to test with pass and error counts.**

## 2. Select a test mode (Functional, Connection, Exclusive, or Online).

### *About Test modes:*

- Connection test mode—provides a low-stress, quick testing of the availability and connectivity of selected devices. These tests are non-intrusive, meaning they will release the devices after a quick test and they will not place a heavy load on system activity.
- Functional test mode—provides more robust testing of your system and devices. It will use your system resources for thorough testing and it assumes that no other applications are running.
- Exclusive test mode—enables users to perform tests that must be performed exclusively, and which allow no other tests or applications to be running concurrently.
- Online test mode—enables users to perform sunvts testing while other customer applications are running concurrently.

### *How to do it:*

- a. Use the Tab key to move to the Control panel (if needed).
- b. Use the Arrow keys to highlight `test_mode` and press Return.

The `test_mode` menu is displayed.

```
test_mode tests_s
ds
*** functional
    connectivity
    exclusive
.]_e[FWD: ^F][DONE: E*]
```

Note: Press Control F (^F) to scroll to the next screen of the `test_mode` menu to display the Online test mode.

```
test_mode tests_s
ds
*** online
.]_e[FWD: ^F][DONE: E*]
```

FIGURE 5-3 `test_mode` Menu (TTY)

- c. Use the arrow keys to highlight the `functional`, `connectivity`, `exclusive`, or `online` test mode and press Return.
- d. Use the arrow key to highlight OK in the Confirmation dialog box and press return.

---

**Note** – You can exit any menu, at any time with the Esc key.

---

### 3. Specify the display mode (logical or physical).

*About display modes:*

- **Logical Mapping**—organizes the devices, according to their function, into groups. For example, SCSI disk, SCSI tape, and SCSI CD-ROM drives are placed in the SCSI-Devices group. You can focus your testing on a specific device, device group, or on all of the groups on the system.
- **Physical Mapping**—shows the exact location of each device on the system in relation to how the devices are connected to each other. If you are testing a single-board type system, each device is shown under the system board. Multi-board type systems show each device under the board (for example, board0, board1, and so on) to which the device is attached. For example, if you have multiple disk drives connected to different disk interfaces, each disk would be displayed under the respective interface. From Physical Mapping, you can determine the actual location of each device. When possible, the board number and controller type for the device are also displayed.

*How to do it:*

#### a. Use the arrow keys to highlight `grouping` and press **Return**.

The `grouping` menu is displayed.

```
grouping set_options 0
*****
logical
physical
Sys>**[DONE;ESC]**  status
                        Tot
                        Elapsed_time:00
```

**FIGURE 5-4** `grouping` Menu (TTY)

#### b. Use the arrow keys to highlight `logical` or `physical` and press **Return**.

### 4. Enable Intervention mode (if needed).

*About Intervention mode:*

Any device that requires your *intervention* before the test can run (such as loading media for tape, CD-ROM, and diskette tests) also requires that you notify SunVTS that you have performed this action by enabling Intervention mode. You cannot select any intervention mode devices until you enable this mode.

*How to do it:*

- a. Prepare the device for testing (see “Preparing Devices for Testing” on page 3-4).
  - b. Use the Tab key to move to the Control panel.
  - c. Use the arrow keys to highlight `intervention` and press return.
  - d. Use the arrow keys to highlight `Enable` and press Return.
5. Select the device groups that you want to test.

*About device groups:*

An asterisk next to a device group in the `Test_Groups` panel indicates that the devices in that group are *selected*. Depending on the devices on your system, and the test mode that you select, SunVTS selects certain devices by default. You can change the selected devices.

*How to do it:*

- a. Use the Tab key to move to the Control panel.
  - b. Use the arrow keys to highlight `test_select` and press Return.
  - c. Use the arrow keys to highlight one of the following:
    - `Default`—selects the SunVTS default selected devices
    - `None`—deselects all devices
    - `All`—selects all devices
    - `Custom`—indicates that you intend to select your own set of devices to test.
6. Select specific devices to test.

*About specific devices:*

In the previous step, you selected sets of devices for testing by choosing device groups. You can further refine your device selection by selecting or deselecting specific devices.

*How to do it:*

- a. Use the Tab key to move to the `Test_Groups` panel.
- b. Use the arrow keys to highlight the device group for which you plan to select or deselect individual devices, and press return.

The individual devices within that group are displayed in a list.



```
*****Processor(s)*****
[*] cpu-unit0(fputest)
[ ] cpu-unit0(cputest)
[ ] system(systemest)

*****[DONE;ESC]*****
```

FIGURE 5-5 Devices Within a Group (TTY)

- c. Use the arrow keys to highlight the [ ] (brackets) next to the device that you want to select or deselect.
- d. Press the space bar to toggle the asterisk.
- e. Press Return to close the group.

Notice that the Status panel displays (or removes) the devices as you select (or deselect) them.

**7. Modify the system-level test options (if desired).**

*About test options:*

SunVTS is ready to test each device once you select it. You can, however, modify test options that alter the way tests run to suit your testing needs. There are three levels where you control test options:

- System-level options—control the testing attributes of *all* devices. This is the highest level, applying all your options globally. When you modify option values at this level, the option settings are propagated down to the group level and device level options.

- Group-level options—control the testing attributes of all the devices in the group. Settings made at this level propagate down, setting all the options in the group.
- Device-level options—control the testing of a single, specific device.

The order that you apply the system-level, group-level, and device-level options is important. Start by assigning the system-level options, then assign the group-level options, and finish with the device-level options. Otherwise, if you make lower-level options, then higher-level options, the higher-level options will propagate down and undo your lower-level options. An alternative to this approach is to use locks and overrides to protect lower-level options. See “Preserving Test Options With Locks and Overrides” on page 5-19.

*How to do it:*

**a. Tab to the Control panel.**

**b. Use the arrow keys to highlight `set_options` and press return.**

The `set_options` menu is displayed.

```

-Hostname: doodle2~~~~~Mod
reset          quit
set_options    option_fil
-----
|*****|
| Thresholds |
| Notify     |
| Schedule   |
| Test_Execution |
| Advanced   |
|*****|
|*** [DONE:ESC]***|

```

**FIGURE 5-6** `set_options` Menu (TTY)

**c. Use the arrow keys to select one of the following menus, and press return:**

- Thresholds
- Notify
- Schedule
- Test\_execution
- Advanced

---

**Note** – For descriptions of these menus see “SunVTS Dialog Boxes” on page A-8.

---

**d. Use the arrow keys and numeric keys to set the options values.**

**e. Use the arrow keys to highlight `Apply` and press Return.**

## 8. Modify group-level and device-level options (if desired).

*About group-level and device-level options:*

Group-level and device-level options are similar to the system-level testing attributes, but instead pertain to a specific device (or device group) only. You modify these options by changing values in the following dialog boxes:

- Test\_Parameters
- Test\_Execution
- Advanced
- Schedule (group-level only)

*How to do it:*

**a. Tab to the Test\_Groups panel.**

**b. Use the arrow keys to highlight the device group and press Return.**

At this point you can choose to access the group-level option menu instead by highlighting `Option` (instead of the device-level) and press return. Then go to Step d. Otherwise, the following steps will direct you to modify device-level options.

**c. Use the arrow keys to select a device from the device list and press Return.**

The `Test_Options` menu is displayed.

```
Processor(s)
[*] cpu-unit0(fputest)
[ ] cpu-unit0(cputest)
[ ] system(systest)
  ***Test_Options***
  [Test_Parameters]
  Test_Execution
  Advanced
  ****[DONE:ESC]****
```

**FIGURE 5-7** Test\_Options Menu (TTY)

**d. Select one of the option dialog boxes (Test\_Parameters, Test\_Execution, or Advanced) and press return.**

---

**Note** – For descriptions of these menus, see “SunVTS Dialog Boxes” on page A-8.

---

**e. Use the arrow and numeric keys to modify the options in the dialog box.**

**f. Use the arrow keys to highlight apply and press return.**

If you selected the `Test_Parameters` dialog box, you will be prompted to apply your changes to one of the following:

- `Within_Instance`—applies the option settings to this test instance (test instance 1 for example) for this device with `Apply`, or to all the same instances within the same device-group with `Apply_to_group`, or to all the same instances for all like devices with `Apply_to_All`.
- `Across_All_Instances`—applies the option settings to all test instances (for example, instance 1 and 2) to this device with `Apply`, or to all the instances for all the devices in the group with `Apply_to_group`, or to all the instances for all like devices with `Apply_to_all`.

```

Apply  Reset
*****Apply Options*****
Within Instance:
  Apply  Apply_To_Group  Apply_To_All
Across Instances:
  Apply  Apply_To_Group  Apply_To_All
*****

```

**FIGURE 5-8** Apply Menu Choices (TTY)

The test session configuration is complete. Continue to the next procedure to run the test session.

## ▼ To Run, Monitor, and Stop a Test Session

### 1. Start the test session.

*How to do it:*

- a. Use the **tab** key to move to the **Control** panel.
- b. Use the **arrow** keys to highlight `start` and press **return**.  
The `start` menu is displayed.
- c. With `start` highlighted (the default) press **return**.

The test session begins.

Depending on how you configured the options in the `Test_Execution` menus, testing will continue until:

- A test error is detected (this is the default behavior). If you enter a value for `Run_On_Error`, SunVTS continues to run until the specified number of errors is reached.
- Testing has reached the `Max_Passes` value. The default is to run indefinite test passes.
- Testing has reached the `Max_Time` value. The default is to run indefinitely.
- You select `Stop` in the Control panel.

### 2. Monitor the overall test session status by watching the Status panel.

The Status panel indicates that a test session is running by displaying *testing* for the `System_status`. The `System_passes`, `Total_errors`, and `elapsed_time` will increment.

### 3. Monitor specific tests by watching the Status panel.

Watch for:

- An asterisk—displayed next to each device indicating which device is currently under test. Depending on how you configured SunVTS (with the System Concurrency option), single or multiple devices are tested concurrently.
- Pass and Error columns—Indicates the state of each device test.

#### 4. Check for test messages by watching the Message console panel.

The Message console panel displays test messages under the following circumstances:

- When a test error occurs
- If you enable trace mode
- If you enable verbose mode

Use the up and down arrow keys to scroll through messages in this panel.

#### 5. Stop the test session (if desired).

*How to do it:*

- a. **Use the Tab key to move to the Control panel.**
- b. **If stop is not highlighted, use the arrow keys to highlight it, and press Return.**  
The test session displays a confirmation menu.
- c. **Press Esc to dismiss the confirmation menu.**  
The test session stops.

## ▼ To Review and Reset Test Session Results

### 1. Review the logs.

*About logs:*

SunVTS provides access to four log files:

- SunVTS test error log—contains time-stamped SunVTS test error messages. The log file pathname is `/var/opt/SUNWvts/logs/sunvts.err`. This file is not created until a SunVTS test failure occurs.
- SunVTS kernel error log—contains time-stamped SunVTS kernel and SunVTS probe errors. SunVTS kernel errors are errors that relate to running SunVTS, and not to testing of devices. The log file pathname is `/var/opt/SUNWvts/logs/vtsk.err`. This file is not created until SunVTS reports a SunVTS kernel error.
- SunVTS information log—contains informative messages that are generated when you start and stop SunVTS test sessions. The log file pathname is `/var/opt/SUNWvts/logs/sunvts.info`. This file is not created until a SunVTS test session runs.

- Solaris system message log—is a log of all the general Solaris events logged by `syslogd`. The pathname of this log file is `/var/adm/messages`.

*How to do it:*

- a. Use the tab keys to move to the Control panel.**
- b. Use the arrow keys to highlight `log_files`.**  
The `log_files` menu is displayed.
- c. Use the arrow keys to select one of the three log files and press return.**
- d. Use the arrow keys to select one of the following:**
  - `Display`—the content of the log file is displayed.

---

**Note** – SunVTS uses your default editor to display the log file. In many cases, the default editor is `vi` and you can use the standard `vi` commands to view the content. However, the file is opened in read-only mode, so you cannot do anything to alter the content. Use the `vi` command `:q` to quit and return to the main TTY window.

---

- `Remove`—the content of the log file is deleted.
- `Print`—the print menu is displayed. Specify the name of the printer in this menu, highlight `Apply`, and press return. The content of the log file is sent to the printer.

---

**Note** – Take the length of the log file into consideration before you print it.

---

- e. Press `<Esc>` to close the `log_file` menu.**

## 2. Interpret SunVTS messages

There are many messages for each of the many tests that run under SunVTS. For this reason, it is difficult to document the implication of each message that you might see. Most messages contain all the explanatory text that is possible for each event that is encountered. Some messages are information messages (`INFO`, `VERBOSE`,

WARNING) which do not represent an error, while some messages are error messages (ERROR and FATAL), indicating that the test detected a failure. This section explains generic test message information.

SunVTS information message example:

```
04/24/00 17:19:47 systemA SunVTS5.1: VTSID 34 disktest.VERBOSE
c0t0d0: "number of blocks 3629760"
```

SunVTS error message example:

```
05/02/00 10:49:43 systemA SunVTS5.1: VTSID 8040 disktest.
FATAL diskette: "Failed get_volmgr_name()"
Probable_Cause(s):
(1)No floppy disk in drive
Recommended_Action(s):
(1)Check the floppy drive
```

Whenever possible, the error message supplies a message type (described in the following table), followed by the message text, one or more probable causes, and one or more recommended corrective actions.

The following table describes the types of messages you might encounter. All messages are displayed in the Message panel, and most are logged in one of the SunVTS logs (Info or Error).

**TABLE 5-3** Message Types

Message Type	Log file	Description
INFO	Info log	Displayed when a non-error test event occurs.
ERROR	Error and Info logs	Displayed when a test detects an error, often an indication of a problem with a specific function or feature of the device under test.
FATAL	Error and Info logs	Displayed when a test detects a severe error that might cause the test to exit, such as a failure to open a device. These errors are indicative of hardware failures.
VERBOSE	Not logged	Test progression messages that are displayed when the verbose feature is enabled.
WARNING	Info log	Displayed when something affects the test that is not an error, such as a device that is busy.

You can write scripts that watch for certain SunVTS events as they are logged in the log files, and which trigger an action when a particular message string occurs. If you



plan to do this, read the information about the message format. See TABLE 4-2.

---

**Note** – As of SunVTS 5.0, the `VTS_OLD_MSG` variable is no longer supported. It was used to display test messages in a pre-SunVTS 4.0 format, usually because a script relies on the older format. Now your scripts must accept the current SunVTS message format.

---

### 3. Reset the test session results (if desired).

*How to do it:*

**a. Use the tab key to move to the Control panel.**

**b. Use the arrow keys to highlight `reset` and press return.**

The information in the Status panel is reset. This does not reset your test options.

---

## 5.3 Using Additional Features With the SunVTS TTY UI

This section describes additional SunVTS features using the SunVTS TTY user interface:

- “Connecting to Another Host” on page 5-18
- “Using the Email Notification Feature” on page 5-18
- “Controlling the Size of the Log Files” on page 5-19
- “Preserving Test Options With Locks and Overrides” on page 5-19
- “Scaling Your Test Session” on page 5-21
- “Using the Debugging Features” on page 5-22
- “Saving a Test Session Configuration for Repeated Use (Option Files)” on page 5-23
- “Using the Auto-Start Feature” on page 5-25
- “Suspending and Resuming a Test Session” on page 5-26
- “Recording and Replaying a Test Session” on page 5-26

## 5.3.1 Connecting to Another Host

You can connect the SunVTS user interface on your local system to a SunVTS kernel that is running on another system in your network. Once you are connected to the remote system, all of the controls in the TTY UI control the testing of the remote system.

### ▼ To Connect to Another Host

#### 1. Make sure that the SunVTS kernel is running on the remote system.

You can do this in several ways. You can remote login (using `rlogin` or `telnet`) to the remote system and run the `vtsk` command (see “To Start the SunVTS Kernel (`vtsk`)” on page 3-7) to start the kernel.

---

**Note** – On the remote system, you must login (or switch user) as a user that has SunVTS privileges (superuser by default) before you can start the SunVTS kernel (`vtsk`).

---

#### 2. Open the `Connect_to_Host` menu:

a. In the Control panel, highlight `connect_to` and press return.

b. Enter the remote system hostname in the `connect_to_host` field.

c. Highlight `Apply` and press return.

d. Press `<Esc>` when you receive the `connect_to` confirmation box.

The SunVTS UI now controls SunVTS on the remote system. The remote system hostname is displayed at the top of the TTY main menu.

## 5.3.2 Using the Email Notification Feature

You can have SunVTS send test status messages to you using email.

### ▼ To Enable Email Notification

#### 1. In the Control panel, highlight `set_options` and press return.

The `set_options` menu is displayed.

#### 2. Select `Notify` and press return.

#### 3. Complete the notify menu items.

#### 4. Highlight `Apply` and press return.

## 5.3.3 Controlling the Size of the Log Files

The log file sizes of the SunVTS test error log (`/var/opt/SUNWvts/logs/sunvts.err`), the SunVTS kernel error log (`/var/opt/SUNWvts/logs/vtsk.err`) and the information log (`/var/opt/SUNWvts/logs/sunvts.info`) are each limited to a maximum size of 1 Mbyte by default. When the log file reaches the maximum size, the content is moved to a file called *logfilename.backup*. Additional events are added to the main log files. If the log file reaches the maximum size again, the content is moved to the backup file, overwriting the earlier backup file content. Only one backup file is maintained for each log file.

You can modify the log file maximum size specification.

### ▼ To Modify the Log File Size Limits

1. **In the Control panel, highlight `set_options` and press return.**  
The `set_options` menu is displayed.
2. **Highlight Thresholds in the `set_options` menu and press return.**
3. **Enter a value (1-5) in the Max System Log Size field.**
4. **With Apply highlighted, press return.**

## 5.3.4 Preserving Test Options With Locks and Overrides

With locks and overrides you can preserve or override the options you set at the system level, group level, and device level.

### 5.3.4.1 Locks

Normally, when you change an option setting at the system level or group level, the new option setting propagates down to all lower levels. Enabling the lock prevents an option setting made at a higher level from affecting the lower-level option setting.

---

**Note** – Overrides nullify lower-level locks.

---

## ▼ To Set (or Unset) a Lock

- **Open the Test Advanced Options dialog box from the group level or device level you want to lock as follows:**
  - a. **Move to the `Test_Groups` panel.**
  - b. **Highlight the group that has the device for which you plan to set (or unset) a lock, and press return.**

The group devices are displayed.
  - c. **Highlight the device and press return.**

The `Test_Options` menu is displayed.
  - d. **Highlight `Advanced` and press return.**

The Advanced options menu is displayed.
  - e. **Highlight the setting for the Test Lock and press return.**
  - f. **Highlight `enabled (or disabled)` and press return.**
  - g. **Highlight `Apply` in the Advanced menu.**

The lock is set (or unset).

## 5.3.4.2 Overrides

You can use overrides to void the lock protection. Setting the system-level override nullifies *all* locks, and setting a group-level override nullifies all the locks below that group.

## ▼ To Set (or Unset) an Override

1. **Open the Advanced Option dialog box at one of the following levels:**
  - System-level—select `Advanced` from the `set_options` in the Control panel.
  - Group-level—select `Advanced` from `Options` next the to group in the `Test_Groups` panel.
2. **Highlight the Override setting (enable/disable), and press return.**

The enable disable menu is displayed.
3. **Select `enable (or disable)`, and press return.**
4. **Highlight `Apply`, and press return.**

## 5.3.5 Scaling Your Test Session

You can use a combination of options to scale testing intensity to meet your diagnostic needs. For example, you can modify test options so that each test instance runs simultaneously, thus increasing the stress level for a single or multiprocessor system. The following procedures can be used individually or in combination to scale the test session.

### ▼ To Modify the Number of Test Instances

You can scale your test session by running multiple copies of the same test on your devices. Each copy is called a *test instance*. Each test instance is a separate process of the same test. The number of test instances can be set at the system level, group level, and the device level as follows:

- 1. Open the `Test_Execution` Option menu at one of the following levels:**
  - **System-level**—select `Test_Execution` from the `set_options` menu in the Control panel.
  - **Group-level**—select `Test_Execution` from `Options` next to the group in the `Test_Groups` panel.
  - **Device-level**—select `Test_Execution` from the device in the device option menu.
- 2. Highlight the `Num of Instances` setting, and enter the value you want.**
- 3. Highlight `Apply`, and press return.**

### ▼ To Modify Test Concurrency Options

The test concurrency option sets the number of tests that run at a given time during a test session. You can set this value to one to run one test at a time, or increase the test concurrency value thus increasing the stress level of your test session.

There are two options you can modify in the `Schedule Option` dialog boxes (at the system level and group level) to scale your test session:

- **System Concurrency**—sets the maximum number of tests that run simultaneously at the system level. It overrides the `Group Concurrency` option.
  - **Group Concurrency**—sets the maximum number of tests that run simultaneously in the same group. This option is available at the system level and the group level.
- 1. Open the `Schedule Option` menu at one of the following levels:**
    - **System-level**—select `Schedule` from the `set_options` menu in the Control panel.
    - **Group-level**—select `Schedule` from `Options` next to the group in the `Test_Groups` panel.

2. **Highlight the concurrency setting, and enter the value you want.**
3. **Highlight `Apply`, and press return.**

### ▼ To Bind a Test to a Processor With the Processor Affinity Option (for Multiprocessor Systems)

By default, the Solaris kernel assigns each test instance to whichever processor is available at the time. On multiprocessor systems, you can bind (assign) a test instance to a particular processor as follows:

1. **Open the `Test_Execution` Option menu at one of the following levels:**
  - System-level—select `Test_Execution` from the `set_options` menu in the Control panel.
  - Group-level—select `Test_Execution` from `Options` next to the group in the `Test_Groups` panel.
  - Device-level—select `Test_Execution` from the device in the device option menu.

---

**Note** – The Processor Affinity field is only displayed on multiprocessor systems.

---

---

**Note** – The Processor Affinity option is not available for processor tests such as `cputest` and `fpctest`. These tests are associated with each individual processor on your system and you cannot bind these tests to a different processor.

---

2. **Highlight the Processor Affinity setting, and enter the value you want.**

By default, the test instances are not bound to any processor. You can either keep the test instances unbound or select a specific processor to bind or select the option for “sequential” binding.

Selecting the “sequential” option binds the test instance(s) to different processors in the system as the test instances are created. Even if there is a higher number of processors than instances, the sequential binding will eventually bind the test instances to all the processors in subsequent test passes.

3. **Highlight `Apply`, and press return.**

## 5.3.6 Using the Debugging Features

You can enable certain options that cause the SunVTS test session to output more test data than when you run the session with the default option values. The following procedures describe how to enable these features.

## ▼ To Enable (or Disable) Debugging Options

1. **Open the `Test_Execution` Option menu at one of the following levels:**
  - System-level—select `Test_Execution` from the `set_options` menu in the Control panel.
  - Group-level—select `Test_Execution` from `Options` next to the group in the `Test_Groups` panel.
  - Device-level—select `Test_Execution` from the device in the device option menu.
2. **Enable (or disable) one or more of the following options:**
  - `Verbose`—displays verbose messages indicating when the test is starting and stopping during the test session.
  - `Core File`—when enabled, creates a core file when a test session results in a core dump. The core file is called `sunvts_install_dir/bin/core.testname.xxxxxx`, where `testname` is the test that dumped the core, and `xxxxxx` is a character string generated by the system in order to make the file name unique.
3. **Highlight `Apply`, and press return.**

### 5.3.7 Saving a Test Session Configuration for Repeated Use (Option Files)

You can use the Option Files feature to save the current set of selected devices and test options for reuse. This is a convenient feature when you plan to use the same test session configuration repeatedly.

The configuration information is saved in a file name of your choice in the `/var/opt/SUNWvts/options` directory.

You can save the option file in host-dependent format or host-independent generic option format. If the option file is stored in host-dependent format, it can be used on other hosts only if the device configurations match. Alternately, the host-independent generic option format is designed to work on multiple hosts with different device configurations. The generic option format may save one set of options for one test which can then be applied to all devices (irrespective of the number of devices) for that test in that machine or other machines. Note that this feature is useful when the number of devices of a specific test is machine-dependent.

While saving the configuration in host-independent generic option format, select at most, one device per device type. If multiple devices of similar type have been selected, any one device will be chosen automatically for being saved. When the same option file is loaded, all devices of that device type will be selected and configured with the same test options.

The host-independent generic option file could be saved for either normal load or cumulative load. A cumulative generic option file could be used to do incremental configuration by loading one after another. Loading a cumulative option file does not deselect already selected test devices. Alternately, loading a non-cumulative option file first deselects all tests, then only selects the devices that are defined in the option file.

The host-independent generic option file save option also allows you to save the total number of instances. If this option is chosen, the same number of instances will be forced during load time.

The option file contains information about system level options and test level options. Option file creation is primarily intended to be done from a GUI. However, you can edit option files with any text editor to accomodate specific needs, if needed. See Appendix C for information on editing generic option files.

## ▼ To Create an Option File

- 1. Configure SunVTS for the test session that you want to save.**

See “To Configure SunVTS for a Test Session” on page 5-4.

- 2. Highlight `option_files` in the Control panel, and press return.**

The `Option Files` menu box is displayed.

- 3. Select an option file name in the `Option File` field.**

- 4. Select Host Dependent or Generic.**

For generic options, select Enable/Disable Cumulative option, and select Enable/Disable Save Number of Instances.

- 5. Highlight `Store`, and press return.**

- 6. Repeat Step 1 through Step 5 for each test session configuration you want to save.**

## ▼ To Load an Option File

---

**Note** – Loading an option file that was created on another system is permitted, but you must make sure that the configuration is valid for the system you plan to test.

---

---

**Note** – Do not load an option file that was created in a 64-bit environment into a 32-bit environment.

---

- 1. Highlight `option_files` in the Control panel, and press return.**

The `Option Files` menu box is displayed.



**2. Highlight `List` and press return.**

The list of available option files is displayed.

**3. Highlight an option file from the list, and press return.**

**4. Highlight `Load`, and press return.**

The test session configuration is loaded into SunVTS. You can use this configuration, or modify it before you start the test session.

▼ **To Remove an Option File**

**1. Highlight `option_files` in the Control panel, and press return.**

The `Option Files` menu box is displayed.

**2. Highlight `List` and press return.**

The list of available option files are displayed.

**3. Highlight the option file from the list that you want to delete, and press return.**

**4. Highlight `Remove`, and press return.**

## 5.3.8 Using the Auto-Start Feature

You can use the auto-start feature, along with saved option files, to simplify the process of configuring a test session and running it. When the auto-start feature is turned on in an option file, SunVTS automatically starts the test session when the option file is invoked, eliminating the need to select the start button.

▼ **To Configure SunVTS to Use the Auto-Start Feature**

**1. Highlight `set_options` from the Control panel, and press return.**

The `set_options` menu is displayed.

**2. Highlight `Schedule` and press return.**

The `Schedule` menu is displayed.

**3. Enable the `Auto Start` setting in the `Schedule` menu.**

**4. Highlight `Apply` and press return.**

**5. Create an option file as described in “To Create an Option File” on page 5-24.**

The option file that you create is configured to use the auto-start feature.

## ▼ To Use the Auto-Start Feature

1. **Highlight** `quit UI` and `Kernel` **from the** `quit` **menu in the Control panel.**  
You must quit SunVTS and restart it for the auto-start feature to work.
2. **Use the following command to restart SunVTS from the command-line, specifying the Option File that you created:**

```
# /opt/SUNWvts/bin/sunvts -to option_file
```

SunVTS displays the TTY main window and automatically starts the test session.

## 5.3.9 Suspending and Resuming a Test Session

You can suspend a test session. For example, you might want to look at messages on the Console panel that have scrolled out of view, or you may want to view and print a log file.

### ▼ To Suspend and Resume a Test Session

1. **While the test session is running, highlight** `suspend` **in the Control panel, and press return.**  
“Suspended” is displayed in the Status panel, and the test session pauses until you resume it.
2. **While the test session is suspended, highlight** `resume` **in the Control panel, and press return.**  
“Testing” is displayed in the Status panel, and the SunVTS kernel resumes the suspended test session.

## 5.3.10 Recording and Replaying a Test Session

You can use the Record and Replay feature to record a SunVTS test session. Only one test session recording is saved at a time.

The events are recorded in a file called `/var/opt/SUNWvts/vts_replay_file`.

Once a test session is recorded, you can use the recorded events to drive the SunVTS kernel so it reproduces the recorded sequence of events at a later time.

---

**Note** – The Record and Replay feature closely reproduces the sequence of events, but it cannot reproduce the time periods of these events because the execution times vary from one run to another.

---

## ▼ To Record and Replay a Test Session

- 1. Configure SunVTS for the test session that you want to run.**

See “To Configure SunVTS for a Test Session” on page 5-4.

- 2. Highlight `start` in the Control panel and press return.**

The start menu is displayed.

- 3. Highlight `Start` with `Record` and press return.**

The test session runs, and the events are recorded. When testing stops, the recorded session is available to replay.

- 4. Highlight `start` in the Control panel and press return.**

The start menu is displayed.

- 5. Highlight `Replay` and press return.**

The kernel reruns the same tests, with the same configuration that you specified.

---

**Note** – During the replay, the kernel is actually rerunning the tests, not just re-displaying the recorded session.

---

---

## 5.4 Scheduling Test Sequences With the Deterministic Scheduler

SunVTS TTY UI provides a Deterministic Scheduler that allows you to determine what tests run, and in what order. If you do not use this feature, selected tests run in an order that is dynamically determined by the SunVTS kernel.

### 5.4.1 Deterministic Scheduler Overview

The Deterministic Scheduler is available as *dsched Menu* in the Control Panel in the SunVTS TTY UI.

---

**Note** – To select an item in the menu, highlight the item using the arrow keys, then press return. Use the <Esc> key to exit a menu, or

---

The following list provides an overview of the steps you perform to start, run, and quit the Deterministic Scheduler feature. Detailed instructions follow.

1. Start SunVTS with the SunVTS TTY user interface as described in Chapter 3.
2. Start the Deterministic Scheduler.
3. Create one or more *tasks*, using the Task Manager menu. A task is a SunVTS test session configuration (one or more selected SunVTS tests, test options, and specific test mode).
4. Create the *sequence*, using the Sequence Manager menu. A sequence is a group of tasks specified to run in a certain order.
5. Define the number of times (*loops*) that sequence runs, using the Sequence Runner menu.
6. Start the sequence.
7. When the sequence is done, reset the Deterministic Scheduler if desired.
8. Quit the Deterministic Scheduler from the main *Dsched Menu*.

## 5.4.2 Getting Started With the Deterministic Scheduler

### ▼ To Start the Deterministic Scheduler

1. Use arrow keys to highlight *dsched* (Deterministic Scheduler) in the control panel and press return.

The Deterministic Scheduler menu is displayed.

```
****DSched Menu****
[ Start DS
  <Show DS>
  <Quit DS Options>
  **** [DONE: ESC] **** ]
```

FIGURE 5-9 DSched Menu (TTY)

2. **Start the Deterministic Scheduler by highlighting *Start DS* and pressing return.**

The DSched Control menu is displayed and the Deterministic Scheduler feature is invoked.

```
-Test_Groups-----
*****
Task Manager
Seq Manager
Seq Runner
*** [DONE:ESC] ***
```

FIGURE 5-10 DSched Control Menu (TTY)

3. **Press <Esc> to return to the main SunVTS TTY window.**

You are returned to the SunVTS main window in preparation to configure tasks.

## ▼ To Create a Task

1. **In the *Test Groups* panel, configure a test session (select one or more devices, test options, and your desired test mode).**

If you assign more than one test to a task, the order of the tests within this task cannot be predicted. If you want to control the order of every test, assign only one test for each task.

---

**Note** – When the Deterministic Scheduler is started, it changes the default values of Max System Passes from 0 (run *indefinitely*) to 1 (run *one* pass). This is necessary so that each task only runs selected tests once and proceeds to run subsequent tasks in the sequence.

---

For details about configuring a test session, refer to “To Configure SunVTS for a Test Session” on page 5-4.

2. **Use arrow keys to highlight *dsched* (Deterministic Scheduler) in the control panel and press return.**
3. **Highlight *Show DS* and press return.**  
The DSched Control menu is displayed (FIGURE 5-10).
4. **Display the Task Manager menu by selecting *Task Manager*, and press return.**  
The *Task Manager* menu is displayed.

```

*****Task Manager*****
Task File:
Load Store Remove Quit List...
*****

```

FIGURE 5-11 Task Manager Window (TTY)

5. In the *Task Manager* menu, enter a task name in the *Task File* text field.
6. Select *Store* and press return to save the task.  
Your task is created with the test configuration you specified in Step 1.
7. Repeat this procedure as needed to create the all the tasks you plan to use in a sequence.

### ▼ To View Existing Tasks

1. Select *List...* from the *Task Manager* menu (FIGURE 5-11) and press return.  
The *Tasks Available* menu is displayed with the list of Available Tasks. If this is the first time that the Deterministic Scheduler is used, you must create a minimum of one task before any tasks are available to view.
2. To see the test session configuration for a particular task, select the task, and press return.
3. Select *Load* and press return.  
The SunVTS Status panel shows the SunVTS test configuration associated with the selected task.

### ▼ To Modify a Task

1. Select *List...* from the *Task Manager* menu (FIGURE 5-11) and press return.
2. Select the task you plan to modify, and press return.
3. Select *Load* and press return.  
The SunVTS Status panel shows the SunVTS test configuration associated with the selected task.
4. Change the test configuration in the main window.
5. In the DSched Task Manager menu, select *Store*.  
Your changes are saved.

## ▼ To Delete a Task

1. Select *List...* from the *Task Manager* menu (FIGURE 5-11) and press return.
2. Select the task you plan to delete, and press return.
3. Select *Remove* in the *Task Manager* menu, and press return.
4. Select *OK* in the confirmation menu, and press return.

The task is deleted.

## ▼ To Create a Sequence

1. Invoke the *Sequence Manager* menu by selecting the *Sequence Manager* (FIGURE 5-12) in the *DSched Control* menu (FIGURE 5-10).

```
-----Status-----|-----Test_Groups-----
|*****Sequence Manager*****|
|Sequence Name:          .sequence|
|Tasks In Sequence      |
|Tasks Insert/Delete    |
|Store Remove Close List...|
|*****|
```

FIGURE 5-12 Sequence Manager Window (TTY)

2. Enter the name of the sequence file you want to create in the *Sequence Name* text field.

By default, the Deterministic Scheduler displays the text field with a *.sequence* suffix. This filename suffix is not mandatory.

3. Select *Task Insert/Delete* -> *Insert Tasks* in the *Sequence Manager* menu.
4. Select *Insert Tasks* and press return.

The *Tasks Available* menu is displayed.

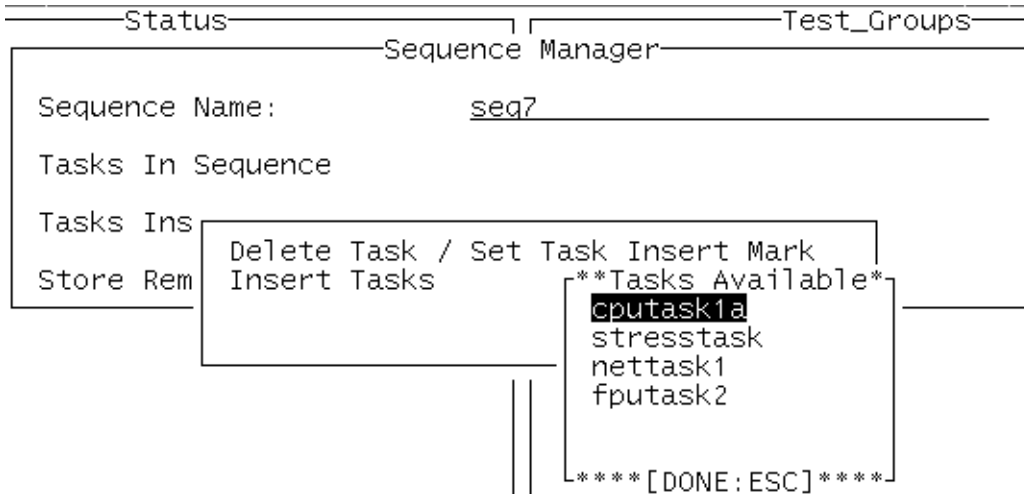


FIGURE 5-13 Tasks Available Menu (TTY)

**5. Select a task from the list and press return.**

This displays a menu with *Insert Task* as the only choice.

**6. Press return.**

The task you selected is inserted into the sequence.

The order in which you add tasks is the order they will be run in the sequence, with the first added task running first, and the last task running last. However, you can insert tasks into specific places in your list of tasks as described later in this section.

**7. Repeat Step 5 and Step 6 to add additional tasks into the sequence.**

---

**Note** – Use the <Esc> key to move back to previous menus.

---

**8. Select *Store* in the Sequence Manager menu (FIGURE 5-13) and press return.**

Your sequence is saved for future use. To run the sequence see “To Run the Sequence” on page 5-35.

---

**Note** – Select *Close* in the Sequence Manager menu to move back to the DSched Control menu.

---



## ▼ To View Tasks in a Sequence

1. **Display the Sequence Manager menu** (accessed from *dsched*►*Show DS*►*Sequence Manager* menu).
2. **Select *List...* in the Sequence Manager menu and press return.**
3. **Highlight the sequence you want to see, and press return.**  
If the list is long, use control-F (forward) and control-B (backwards) to scroll through the sequences.

```
Sequence Manager
Sequence Name:      seq10a
Tasks In Sequence
Tasks Insert/Delete
Store Remove Close List...
*****Sequence Files*****
.sequence cpuSEQ   seq7
seq10a
***** [DONE:ESC] *****
```

FIGURE 5-14 List of Sequences (TTY)

4. **You can view the list of tasks available in this sequence by selecting *Tasks In Sequence* in the *Sequence Manager* menu.**

## ▼ To Remove Tasks From a Sequence

1. **Display the Sequence Manager menu** (accessed from *dsched*►*Show DS*►*Sequence Manager* menu).
2. **Select *list...* in the Sequence Manager menu and press return.**  
The list of sequences is displayed.
3. **Select the sequence you plan to modify and press return.**
4. **Select *Task Insert / Delete* and press return.**  
The *Tasks In Sequence* menu is displayed with a list of tasks in the sequence.
5. **Select *Delete Task/Set Task Insert Mark* and press return.**  
The list of tasks for this sequence is displayed.

6. **Select the task you want to delete and press return.**  
A menu with two choices is displayed.
7. **Select *Delete Task* and press return.**  
The selected task is deleted from the sequence.
8. **Repeat the above steps to remove additional tasks as desired.**
9. **Use the <Esc> key to move back to the Sequence Manager menu.**
10. **Save your changes by selecting *Store* and press return.**
11. **Select *ok* and press return.**
12. **Select `Close` to exit the Sequence Manager menu.**

### ▼ To Insert Tasks in a Sequence

1. **Display the Sequence Manager menu (accessed from *dsched*►*Show DS*►*Sequence Manager* menu).**
2. **Select *list...* in the Sequence Manager menu and press return.**  
The list of sequences is displayed.
3. **Select the sequence you plan to modify and press return.**
4. **Select *Task Insert/Delete Task* and press return.**
5. **Delete *Set Task Insert Mark* and press return.**  
The list of tasks for this sequence is displayed.
6. **Use the arrow keys to select the task that is directly above the place where you want the new task inserted, and press return.**
7. **Select *Set Insert Mark*, and press return.**  
This marks your insertion point. Any task you add after this step, in this sequence, will be inserted below this insertion mark.
8. **Use the <Esc> key to move back to the Insert Task menu.**
9. **Select *Insert Task* and press return.**  
The list of available tasks is displayed.
10. **Select the task you want to add and press return.**
11. **With *Insert Task* highlighted, press return.**  
The task is inserted below the insert mark.
12. **Repeat Step 10 and Step 11 for all the tasks you want to add.**

13. Use the <Esc> key to move back to the Sequence Manager menu.
14. Save your changes by selecting *Store* and pressing return.
15. Select *ok* and press return.
16. Select *Close* to exit the Sequence Manager menu.

### ▼ To Delete a Sequence

1. **Display the Sequence Manager menu** (accessed from *dsched*►*Show DS*►*Sequence Manager* menu).
2. **Select *list...* in the Sequence Manager menu and press return.**  
The list of sequences is displayed.
3. **Highlight the sequence you want to delete, and press return.**
4. **Select *Remove* in the Sequence Manager menu and press return.**
5. **Select *ok* and press return.**  
The selected sequence is deleted from the Deterministic Scheduler.

---

**Note** – The tasks associated with the deleted sequence are not deleted.

---

6. Select *Close* to exit the Sequence Manager menu.

### ▼ To Run the Sequence

1. **Invoke the *Sequence Runner* menu by selecting *Seq Runner* in the *Start DS* menu.**

```

*****Sequence Runner*****
Sequence File:      seq7
List...

Total Loops:[1-10]  1_
Current Loop:      0
Tasks in Sequence:
  fputask2
  fputask2

```

FIGURE 5-15 Sequence Runner Menu (TTY)

2. **Select the sequence that you want to run by selecting *List...* , highlighting the sequence, and press return.**

3. Change the *Total Loop* value to the number of times you want to repeat the sequence of tests.

---

**Note** – The default and minimum number of *Total Loops* value is 1. You can change the *Total Loop* value to the desired value, up to a maximum of 10 loops.

---

4. Select the *Start* button at the bottom of the *Sequence Runner* menu start the sequence.

While testing is underway, you can do any of the following:

- Monitor the test progress in the TTY Status panel. All test results are logged in the SunVTS log files.
- Suspend and resume the sequence by selecting the *Suspend* and *Resume* commands in the *Sequence Runner* menu.
- Close the *Sequence Runner* menu by selecting the *Quit DS UI* from the *Quit DS* menu. Testing will continue, and results will be displayed in the TTY Status panel.

---

**Note** – When the Deterministic Scheduler is running, the selection of *Show DS* displays the *Sequence Runner* menu. Otherwise, this selection causes the *Start DS* menu to display.

---

## ▼ To Reset the Status Panel and Deterministic Scheduler

1. Use the *Reset* in the main window to reset the Status panel and Deterministic Scheduler.

The TTY Status panel test statistics and the Deterministic Scheduler current loop count are reset.

## ▼ To Quit the Deterministic Scheduler

1. To quit the Deterministic Scheduler when you are done, select *dsched*→*Quit DS Options*→*Quit DS*.

## SunVTS Window and Dialog Box Reference

---

This appendix provides descriptions of each SunVTS main window and dialog box. See the following:

SunVTS Main Windows:

- “CDE Main Window” on page A-2
- “TTY Main Window” on page A-7

SunVTS Dialog Boxes:

- “Advanced” on page A-8
- “Auto Configuration” on page A-9
- “Connect to Host” on page A-10
- “DSched” on page A-11
- “Notify” on page A-13
- “Option Files” on page A-14
- “Schedule” on page A-16
- “Test Execution” on page A-17
- “Test Parameter” on page A-19
- “Thresholds” on page A-21
- “Trace Test” on page A-22

# A.1 SunVTS Main Windows

## A.1.1 CDE Main Window

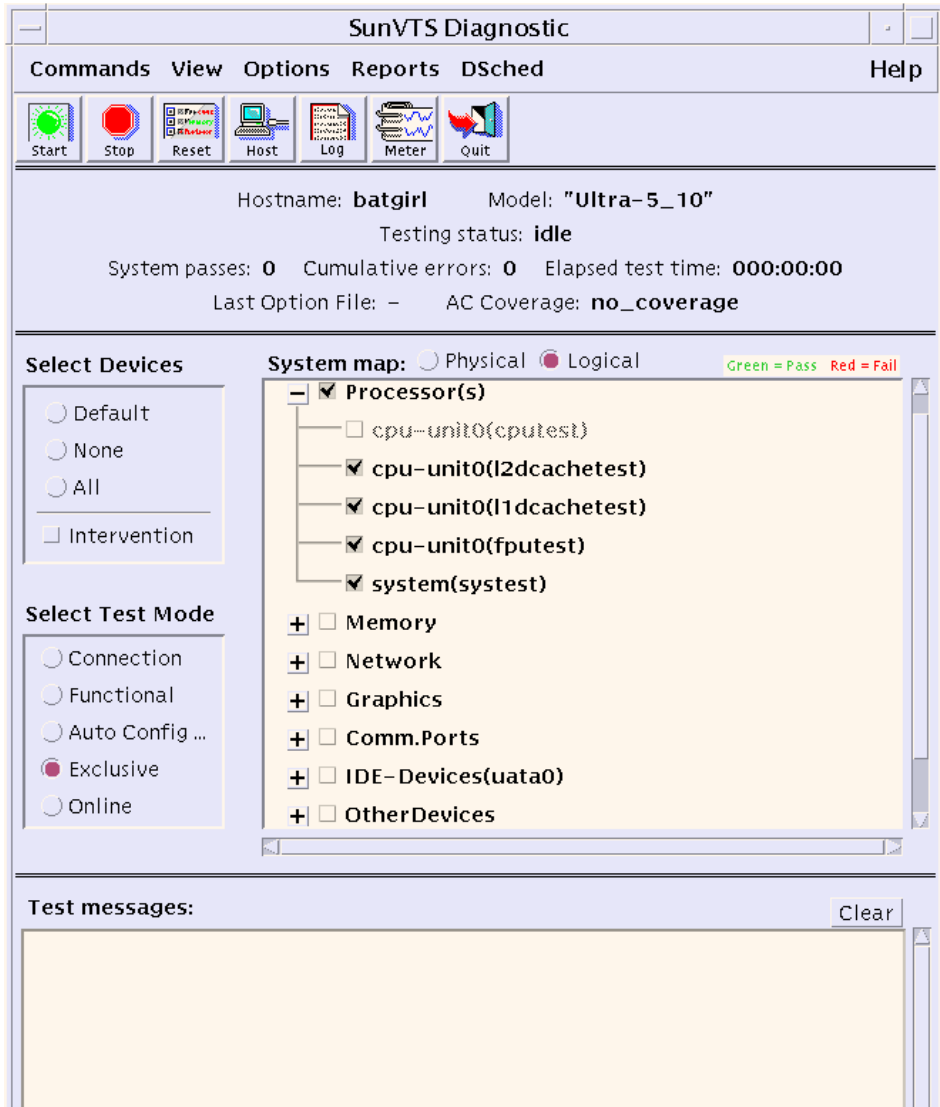


FIGURE A-1 SunVTS CDE Main Window

**TABLE A-1** SunVTS CDE Main Window

Window Item	Description
Menu Bar	<p>Provides access to a variety of SunVTS controls through the following pull-down menus:</p> <p>Commands Menu:</p> <ul style="list-style-type: none"><li>• Start testing: Starts enabled tests, records a test session, or replays a recorded test session</li><li>• Stop: Stops testing</li><li>• Reset: Sets pass/error count to zero; pauses</li><li>• Suspend: Pauses active tests</li><li>• Resume: Resumes testing</li><li>• Connect to Host: Connects to another machine running SunVTS kernel</li><li>• Trace test: traces enabled tests</li><li>• Reprobe system: probes system again</li><li>• Quit SunVTS: quits the following: UI and Kernel, UI only, Kernel only</li></ul> <p>View Menu:</p> <ul style="list-style-type: none"><li>• Open System map</li><li>• Close System map</li></ul> <p>Options Menu: Note - To increase or decrease a numeric value in a SunVTS CDE dialog box, you can either use the up or down arrows, or type a new value in the text box and press Return. Press Apply to apply all dialog box changes. (detailed descriptions of each menu are provided in this chapter):</p> <ul style="list-style-type: none"><li>• Thresholds</li><li>• Notify</li><li>• Schedule</li><li>• Test execution</li><li>• Advanced</li><li>• Option files</li></ul> <p>Reports Menu:</p> <p>Displays, prints, or removes log file information.</p> <p>Help Menu:</p> <p>Shows the SunVTS version</p> <p>Displays the Solaris/SunVTS release table</p>

**TABLE A-1** SunVTS CDE Main Window (*Continued*)

Window Item	Description
	DSched Menu: <ul style="list-style-type: none"><li>• Start DS—starts the Deterministic Scheduler feature and displays the Deterministic Scheduler dialog box.</li><li>• Show DS—displays the Deterministic Scheduler dialog box after you quit the DS UI.</li><li>• Quit Options—provides access to the following quit options:</li><li>• Quit DS UI only—quits the Deterministic Scheduler dialog box and leaves the Deterministic Scheduler running.</li><li>• Quit DS—quits the Deterministic Scheduler dialog box and stops the Deterministic Scheduler.</li></ul>
Quick Access buttons	Provides easy access to the most commonly used SunVTS controls: <ul style="list-style-type: none"><li>• Start—starts the test session.</li><li>• Stop—stops the test session.</li><li>• Reset—sets the pass and error counts to zero, the test colors to black, and the system status information to zero.</li><li>• Host—opens the Connect to Host dialog box.</li><li>• Log—opens the Log Files dialog box.</li><li>• Meter—opens the Performance meter panel.</li><li>• Quit—closes the SunVTS UI and leaves the SunVTS kernel running.</li></ul>



**TABLE A-1** SunVTS CDE Main Window (*Continued*)

Window Item	Description
Status panel	<p>Displays the overall SunVTS testing status as follows:</p> <ul style="list-style-type: none"> <li>• <b>Hostname</b>—indicates the name of the system under test</li> <li>• <b>Model</b>—indicates the model of the system under test</li> <li>• <b>Testing status</b>—provides the following status of SunVTS: <ul style="list-style-type: none"> <li>• <b>ds_idle</b>—the deterministic scheduler is invoked, but is not testing</li> <li>• <b>ds_running</b>—the deterministic scheduler is invoked and is running a sequence</li> <li>• <b>ds_suspended</b>—the deterministic scheduler is invoked and a sequence is suspended</li> <li>• <b>idle</b>—no testing is taking place</li> <li>• <b>replay</b>—a previously recorded test session is displayed</li> <li>• <b>stopping</b>—is momentarily displayed when a test session stops</li> <li>• <b>suspended</b>—a test session is suspended</li> <li>• <b>Testing</b>—a test session is running</li> </ul> </li> <li>• <b>System passes</b>—displays the total number of successful system passes (a system pass is when all tests have been run once).</li> <li>• <b>Cumulative errors</b>—the total number of errors from all tests</li> <li>• <b>Elapsed test time</b>—indicates the total number of hours, minutes, and seconds that elapsed during testing.</li> <li>• <b>Last Option File</b>—displays the name of the last option file that was accessed.</li> <li>• <b>AC Coverage</b>—displays the type of Auto Configuration coverage level that was selected (either <code>confidence</code>, <code>comprehensive</code>, or <code>no_coverage</code> when the AC feature is not in use).</li> </ul>
Select Devices panel	<p>Provides buttons that control the devices that are selected for testing. This panel also provides the button used to enable intervention mode.</p>

**TABLE A-1** SunVTS CDE Main Window (*Continued*)

Window Item	Description
Select mode panel	<p>Provides buttons that control the test mode:</p> <ul style="list-style-type: none"><li>• Connection test mode—provides a low-stress, quick testing of the availability and connectivity of selected devices. These tests are non-intrusive, meaning they will release the devices after a quick test, and they will not place a heavy load on system activity.</li><li>• Functional test mode—provides more robust testing of your system and devices. It will use your system resources for thorough testing and it assumes that no other applications are running.</li><li>• Auto Functional mode—opens the Automatic Configuration dialog box which provides access to the Automatic Configuration feature. The Automatic Configuration feature simplifies the test session configuration steps that you perform by automatically selecting a consistent and reliable set of test options for the platform under test. See “Using the Automatic Configuration Feature” on page 4-18 for instructions on using this feature.</li></ul>
System Map	<p>Displays selectable devices and provides access to control the individual selection and testing attributes for each device. The following items are displayed in the System Map:</p> <p>Collapse/Expand symbols (+, -)—control the way devices are displayed for each group. The + (plus) indicates devices are collapsed (hidden from view). The - (minus) indicates that devices are expanded (you see each device).</p> <ul style="list-style-type: none"><li>• Groups—represents a collection of devices. If you select a group for testing, all devices in that group are tested.</li><li>• Device—is the name of an individual device that is selectable for testing.</li><li>• Test name—is the name of the SunVTS test that is assigned to test the device. The test name is displayed in parentheses.</li><li>• Checkmark box—Specifies whether a device or device group is selected for testing.</li><li>• Colors—Indicate the state of devices; Red=test failure, Green=no failure, Black=device not tested.</li><li>• Asterisk (*)—indicates that the device is currently being tested.</li></ul>
Test Message panel	Displays test session messages.

## A.1.2 TTY Main Window



**FIGURE A-2** SunVTS TTY Main Window

**TABLE A-2** SunVTS TTY Main Window

Window Item	Description
Control panel	Provides access to SunVTS commands and options.
Test_Groups panel	Shows the testable devices (categorized in groups) and provides access to their test options.
Status panel	Displays SunVTS test status information.
Message panel	The area where test messages are displayed. No user input is accepted in this panel.

## A.2 SunVTS Dialog Boxes

### A.2.1 Advanced

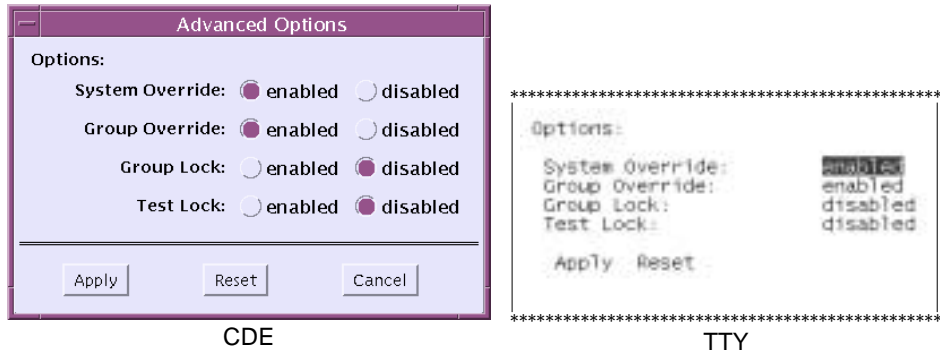
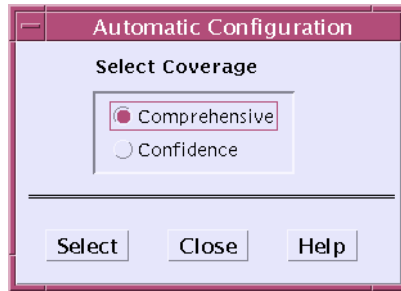


FIGURE A-3 Advanced Dialog Boxes

TABLE A-3 Advanced Dialog Box Descriptions

Item	Description
System Override	Sets all options for all tests to the settings made at the system level. Supersedes the group-specific and device-level option settings in favor of the options set at the global level.
Group Override	Supersedes the device-level test options in favor of the group options. The System Override option supersedes this option.
Group Lock	Protects specific group options from being changed from the options set at the system level (the System Override option supersedes this option).
Test Lock	Protects device-level options from being changed from options set at the group or system level (the System Override and Group Override options supersede this option).

## A.2.2 Auto Configuration



**FIGURE A-4** Auto Configuration Dialog Box

**TABLE A-4** Auto Configuration Dialog Box Descriptions

Item	Description
Comprehensive	Sets all the test options for thorough testing. All applicable functional tests are enabled. The test session validates the full functionality of the system and provides you with assurance that the system is hardware fault free.
Confidence	Sets all the test options such that there is limited functional test coverage as compared to the Comprehensive level of coverage. The Confidence level testing verifies the main functionality of the system in a shorter amount of time.

---

**Note** – The Help button displays a dialog box with Auto Configuration information and instructions.

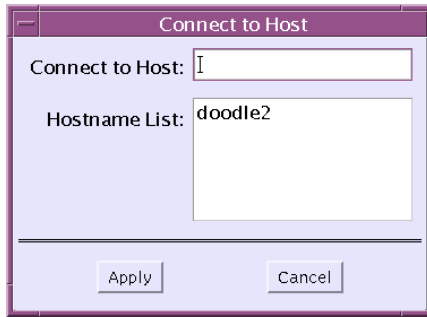
---

---

**Note** – The Auto Configuration feature is only available from the SunVTS CDE UI.

---

## A.2.3 Connect to Host



CDE



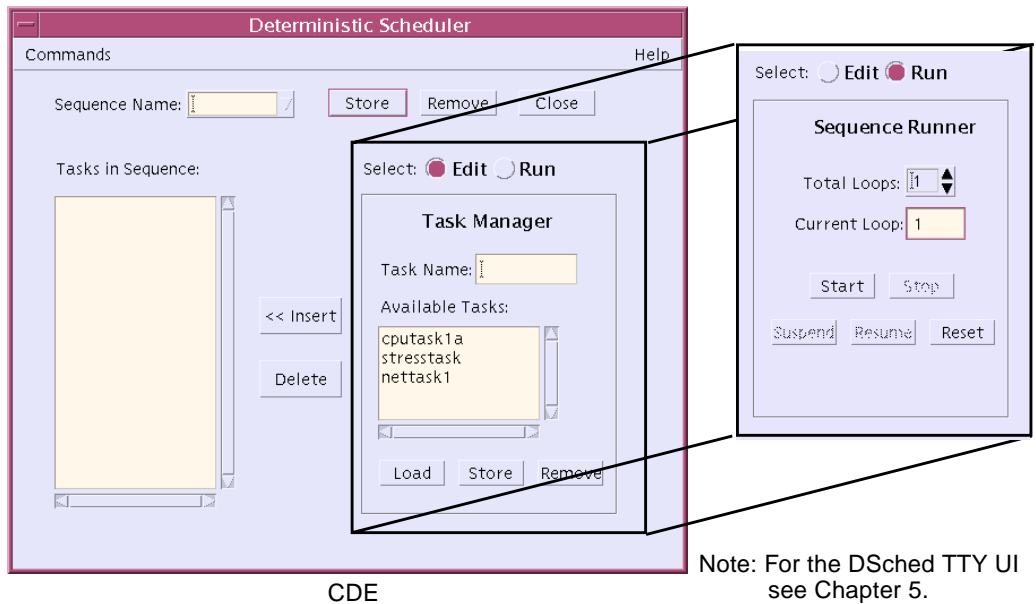
TTY

**FIGURE A-5** Connect to Host Dialog Boxes

**TABLE A-5** Connect to Host Dialog Box Descriptions

Item	Description
Connect to Host	Specifies the hostname of the system to which you want to connect. The host you specify must be running the SunVTS kernel (the same version as the local UI), and you must have permission.
Hostname List	Displays the list of hosts that this UI has connected to, making them available for future connections. This list is cleared when you quit SunVTS.

## A.2.4 DSched



**FIGURE A-6** Deterministic Scheduler Dialog Boxes

**TABLE A-6** Deterministic Scheduler Dialog Box Descriptions

Dialog Box Item	Description
Commands	Provides access to the following Deterministic Scheduler quit commands: Quit UI Only—quits the Deterministic Scheduler dialog box and leaves the Deterministic Scheduler running. Quit DS and UI—quits the Deterministic Scheduler dialog box and stops the Deterministic Scheduler.
Sequence Name	A field used to specify a unique sequence name for a group of tasks that are listed in the Tasks in Sequence area. You can access existing sequences by clicking on the down arrow that is next to this field. The Sequence Name buttons provide the following functions: Store—saves the named sequence for future use. Remove—deletes the named sequence. Close—quits the Deterministic Scheduler dialog box and leaves the Deterministic Scheduler running.
Tasks in Sequence	Lists the tasks in a specific order for the current sequence.

**TABLE A-6** Deterministic Scheduler Dialog Box Descriptions (*Continued*)

Dialog Box Item	Description
Insert and Delete buttons	Inserts and deletes tasks from the Tasks in Sequence area.
Select Edit   Run	Specifies the Deterministic Scheduler panel as follows: Edit—displays the Deterministic Scheduler edit panel. In this panel, the Scheduler is ready for you to create, load, and remove tasks and sequences. Run—displays the Deterministic Scheduler run panel. In this panel, the Scheduler is ready for you to define the number of times (loops) for a sequence to run, start, suspend, resume, stop, and reset the Deterministic Scheduler.
Edit panel	Appears when the Select radio button is set to Edit. The following functions are displayed: Task Name—a field used to specify a task name. A task is a group of tests that are selected in the SunVTS main window. Available Tasks—shows a list of existing tasks. Load—loads the selected task into the SunVTS main window. Store—saves the task. Remove—deletes the selected task from the task list.
Run panel	Appears when the Select button is set to Run. The following functions are displayed: Total Loops—specifies the number of times to run a sequence. Current Loop—displays each loop number as a sequence runs. Start—Starts running the sequence that is specified in the Sequence Name field. Stop—stops the running sequence. Suspend—pauses a running sequence. Resume—resumes a suspended sequence. Reset—resets the SunVTS main window and the sequence dialog box.

---

**Note** – The Help button displays a dialog box with Deterministic Scheduler information and instructions.

---

---

**Note** – The Deterministic Scheduler is also available in the SunVTS TTY UI. The menus for the Deterministic Scheduler TTY version are described in “Scheduling Test Sequences With the Deterministic Scheduler” on page 5-27.

---



## A.2.5 Notify

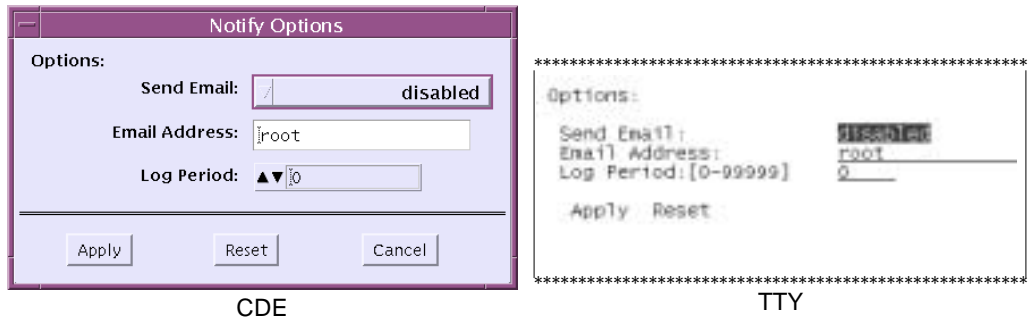
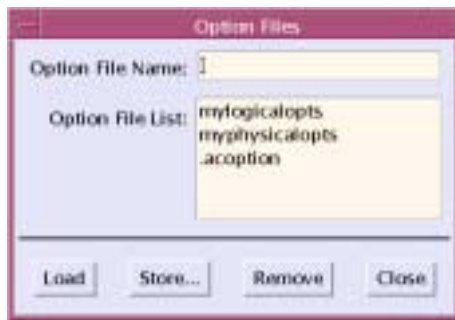


FIGURE A-7 Notify Dialog Boxes

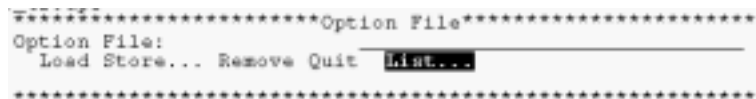
TABLE A-7 Notify Dialog Box Descriptions

Item	Description
Send Email	Determines when and if you want the test status messages sent to you by email. From this menu, you can choose: <ul style="list-style-type: none"> <li>• Disabled—never send email.</li> <li>• Now —send email immediately.</li> <li>• On Error —send email when an error occurs.</li> <li>• Periodically —send email with a time period set in the Log Period option.</li> <li>• On Error &amp; Periodically —send email periodically as well as when an error occurs.</li> </ul>
Email Address	Indicates the email address where the test status messages are sent (the address is <code>root</code> by default).
Log Period	Specifies the time, in minutes, between sending test status email messages.

## A.2.6 Option Files



CDE



TTY

FIGURE A-8 Option Files Dialog Boxes

TABLE A-8 Option Files Dialog Box Descriptions

Item	Description
Option Files	Opens the Option Files dialog box. Use this dialog box to name and store a SunVTS configuration for future use. Later, you can open this dialog box to select and load an option file for quick SunVTS configuration.
Option File List	Displays the list of option files, from which you choose to load, or remove.



CDE

```

*****Store Option File*****
Option File Name: mylogicalopt

Option File Type:
[ X ] Host Dependent
[ ] Generic

-----

For Generic Option Files Only:
[ ] Save For Cumulative Load
[*] Save Number of Instances

Store Cancel
***** [DONE:ESC] *****

```

TTY

**FIGURE A-9** Store Option File Dialog Boxes

**TABLE A-9** Store Option File Dialog Box Descriptions

Item	Description
Option File Name	Specifies the option file selected for storing. Use this dialog box to enter the name of the option file to be stored.
Option File Type	Host Dependent—enables the option file to be used on another host if the device configurations match. Generic— enables saving one set of options for one test that can be applied to all devices (irrespective of the number of devices) for that test in that machine or another machine. Note that the Generic option format is useful when the number of devices of a specific test is machine-dependent.
For Generic Option Files Only	Cumulative—enables incremental configuration by loading one after another; loading a cumulative option file does not deselect already selected test devices. Alternately, loading of non-cumulative option file will first deselect all tests, then only select the devices that are described in the option file. Save Number of Instances—enables the same number of instances to be forced during load time.

## A.2.7 Schedule

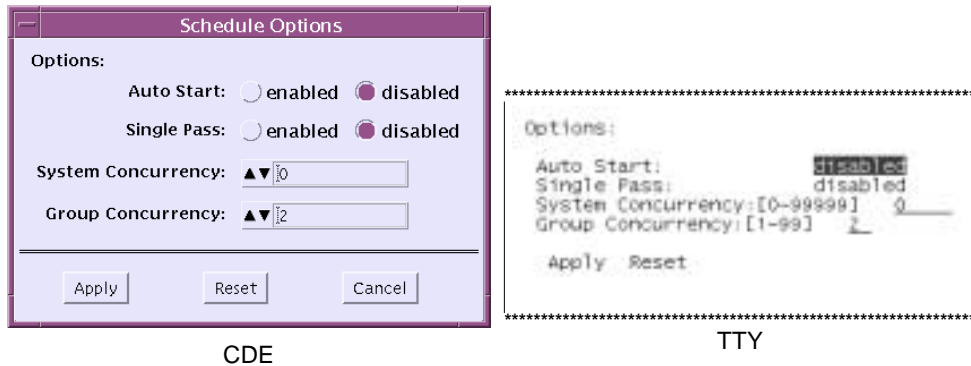


FIGURE A-10 Schedule Dialog Boxes

TABLE A-10 Schedule Dialog Box Descriptions

Item	Description
Auto Start	Eliminates the need to use the Start button to start a SunVTS testing session. To use this option see “Using the Auto-Start Feature” on page 4-28.
Single Pass	Runs only one pass of each selected test.
System Concurrency	Sets the maximum number of test instances that can be run concurrently on the machine being tested.
Group Concurrency	Sets the number of tests you want to run at the same time in the same group.

## A.2.8 Test Execution

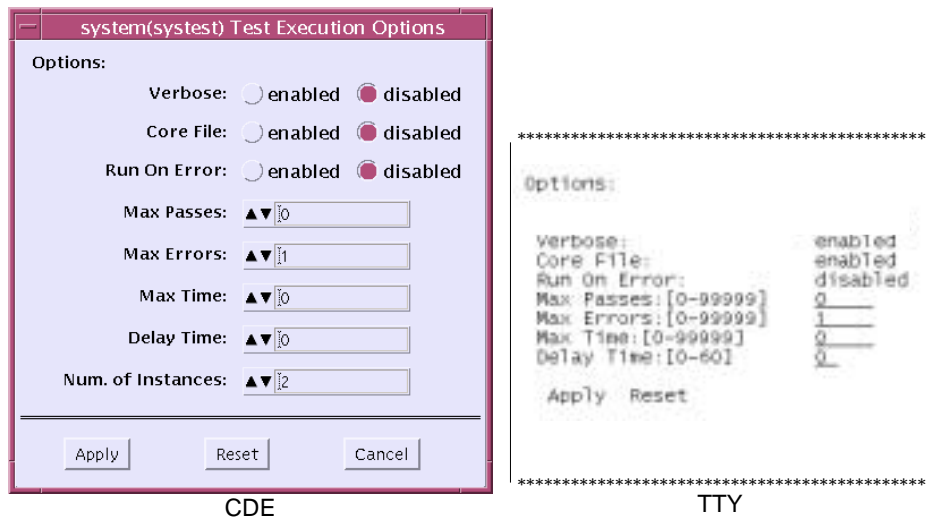


FIGURE A-11 Test Execution Dialog Boxes

TABLE A-11 Test Execution Dialog Box Descriptions

Item	Description
Stress	This option is no longer available as of SunVTS 5.0.
Verbose	Displays verbose messages in the SunVTS messages window.
Core File	When enabled, creates a core file when a test session results in a core dump. When Core File is disabled, a core file is not created. Instead, a message is displayed and logged, indicating the signal that caused the failure. See “Using the Debugging Features” on page 4-25.
Run On Error	When disabled, SunVTS stops testing when an error is encountered. When enabled, SunVTS continues to run until the Max errors value is reached.
Max Passes	Specifies the maximum number of test passes for all selected devices. The default is 0 (zero) which means there is no limit.
Max Errors	(Only enabled when the Run On Error option is enabled) Specifies the maximum number of allowable errors before stopping the test session. The default is 0 (zero) which directs the SunVTS kernel to continue testing in the event of all errors.

**TABLE A-11** Test Execution Dialog Box Descriptions (*Continued*)

Item	Description
Max Time	Specifies the time limit, in minutes, that a test session can run. The default is 0 (zero) which means there is no time limit.
Delay Time	<p>Specifies, in seconds, the amount of time to pause between execution of a test. This option results in differing test behavior based on the level for which this option is specified:</p> <p>System level—the delay is performed between the execution of all selected tests.</p> <p>Group level—the delay is performed between the execution of selected tests in this group.</p> <p>Device level—the delay is performed between the execution of the test for this device.</p> <p>Select a value from 0 to 60. Zero (0), the default, performs no delay. This option offers testing convenience for tests such as graphic device tests, where adding a delay between test execution provides a period of time for you make selections in the SunVTS main window.</p>
Number of Instances	Specifies the number of instances to run each scalable test.

## A.2.9 Test Parameter

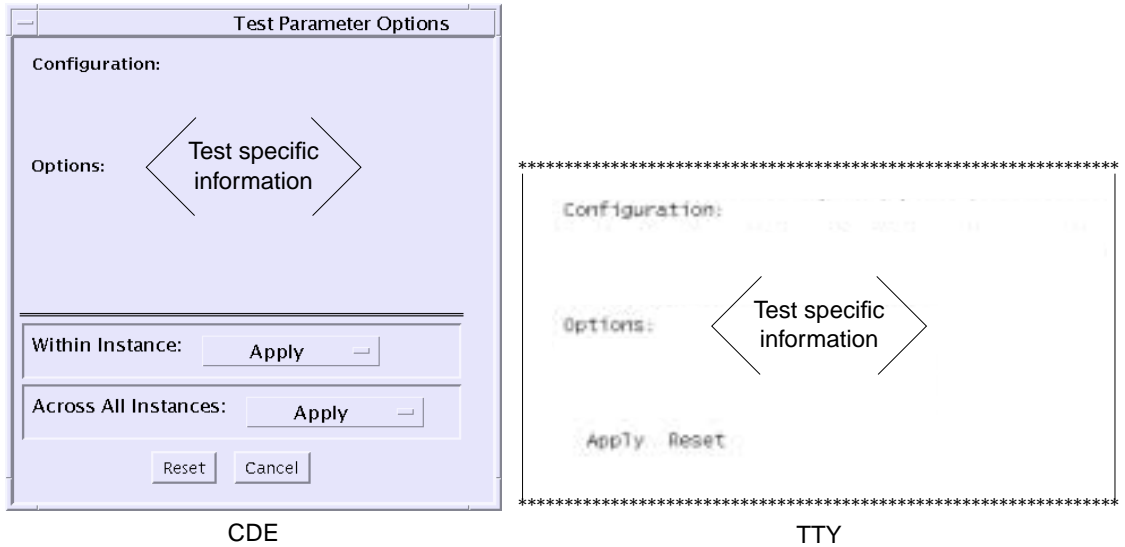


FIGURE A-12 Test Parameter Dialog Boxes

---

**Note** – Except for the lower buttons, this menu is different for each device. For descriptions of each different Test Parameter Options Dialog Box, refer to the *SunVTS Test Reference Manual*.

---

**TABLE A-12** Test Parameter Options Dialog Box Descriptions (Generic Lower Buttons)

Item	Description
Processor Affinity	Only available on multiprocessor systems, used to specify on which processor you want to run all tests. To select the specific processor, click the processor number. If no processor is specified, the operating system randomly distributes testing among all the processors. Note: when you use random distribution, you cannot verify that all processors have been tested.
Within Instance	Provides the means to apply the settings: <ul data-bbox="529 527 1219 673" style="list-style-type: none"><li>• To this device only with the Apply button, or</li><li>• To all devices within this group with the Apply to Group button, or</li><li>• To all devices (of the <i>same device type for all controllers</i>) with the Apply to All button.</li></ul> The option settings are only applied to one instance of the test.
Across All Instances	Provides the means to apply the settings globally: <ul data-bbox="529 762 1219 907" style="list-style-type: none"><li>• To this device only with the Apply button, or</li><li>• To all devices within this group with the Apply to Group button, OR</li><li>• To all devices (of the <i>same device type for all controllers</i>) with the Apply to All button.</li></ul> The option settings are applied to all instances.
Reset	Returns the option values to their default settings and closes the test parameter option menu.
Cancel	Ignores any changes made to option values and closes the test parameter option menu.



## A.2.10 Thresholds

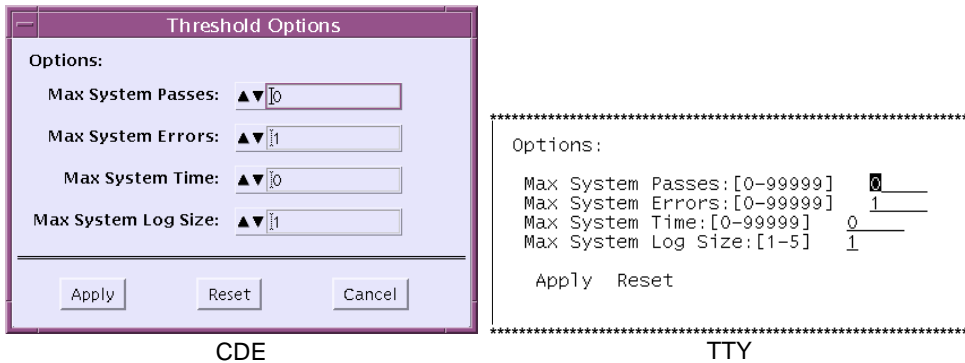
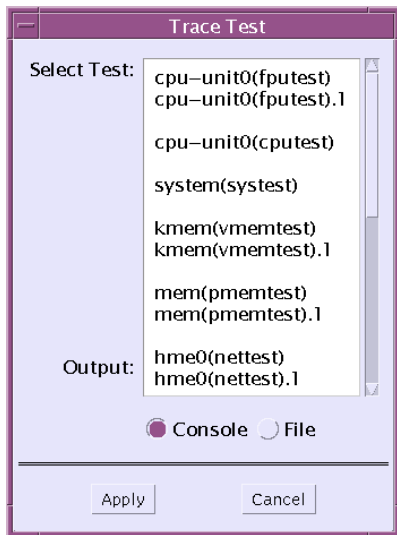


FIGURE A-13 Threshold Dialog Boxes

TABLE A-13 Threshold Dialog Box Descriptions

Item	Description
Max System Passes	Specifies the maximum number of system passes before stopping all tests. The default of 0 (zero) causes the SunVTS kernel to run the tests until you click the Stop button. Note: The definition of a <i>system pass</i> is when all selected tests have run one time.
Max System Errors	Specifies the maximum number of system errors before SunVTS stops all tests. The default is 0 (zero), which causes the SunVTS kernel to continue testing regardless of errors.
Max System Time	Specifies the maximum number of minutes that SunVTS continues testing. The default of 0 (zero) makes the SunVTS kernel run the selected tests until you click the Stop button.
Max System Log Size	Specifies the maximum size of the SunVTS error log file ( <code>/var/opt/SUNWvts/logs/sunvts.err</code> ), the SunVTS kernel error log ( <code>/var/opt/SUNWvts/logs/vtsk.err</code> ) and the info log file ( <code>/var/opt/SUNWvts/logs/sunvts.info</code> ) in Mbytes. The default is 1 Mbyte. Specify an integer from 1 to 5.

## A.2.11 Trace Test



CDE

**FIGURE A-14** Trace Test Dialog Boxes

**TABLE A-14** Trace Test Dialog Box Descriptions

Item	Description
Select Test panel	Displays the tests that are available to select for tracing.
Console/File button	Specifies whether to send the trace messages to the message panel, or to the trace file ( <code>/var/opt/SUNWvts/logs/sunvts.trace</code> ).

**Note** – The Trace mode dialog box is only available from the SunVTS CDE UI.

## Frequently Asked Questions

---

### 1. Which version of SunVTS should I install?

As of SunVTS 5.1, the SunVTS software is compatible with multiple Solaris operating environment software releases for backwards compatibility. The following table lists the Solaris operating environment release to SunVTS version correspondence.

**TABLE B-1** SunVTS Revision to Solaris Release Support Table

<b>Solaris Operating Environment Release</b>	<b>SunOS Version</b>	<b>SunVTS Revision</b>
9, 9 9/02, & 8 2/02	5.9 or 5.8	5.1
9	5.9	5.0
8 2/02	5.8	4.6
8 10/01	5.8	4.5
8 7/01	5.8	4.4
8 4/01	5.8	4.3
8 1/01	5.8	4.2
8 10/00	5.8	4.1
8 6/00	5.8	4.0
8	5.8	4.0
7 11/99	5.7	3.4
7 8/99	5.7	3.3
7 5/99	5.7	3.2
7 3/99	5.7	3.1
7	5.7	3.0

**TABLE B-1 SunVTS Revision to Solaris Release Support Table**

<b>Solaris Operating Environment Release</b>	<b>SunOS Version</b>	<b>SunVTS Revision</b>
2.6 5/98	5.6	2.1.3
2.6 3/98	5.6	2.1.2
2.5.1 11/97	5.5.1	2.1.1

2. **Can I install SunVTS in a directory other than the default (/opt)?**  
Yes. See “Installing and Removing SunVTS” on page 21.
3. **Will installing SunVTS negatively affect any of the existing files on my system?**  
No. SunVTS follows standard packaging conventions, installing cleanly in your environment, and can be removed cleanly with the `pkgrm` command.
4. **Can I install multiple versions of SunVTS on my system.**  
No. Only one installation of SunVTS is supported on a given system.
5. **If I reinstall SunVTS, will I have access to the options files that were created before the reinstall?**  
Yes. The option files are not affected by installation or deinstallation.
6. **Can I install and run SunVTS on my x86 system that is running the Solaris Operating Environment?**  
No. SunVTS is not ported to Solaris on x86. SunVTS is only supported on Sun SPARC/Solaris environments.
7. **Can I run SunVTS directly from the Solaris Supplement CD that contains SunVTS?**  
No. You must install SunVTS as described in “Installing and Removing SunVTS” on page 21 on each system for which you plan to run SunVTS.
8. **Can I run SunVTS from maintenance-mode (single-user level)?**  
No. Your system must be running multi-user mode.
9. **Can I run SunVTS through the Sun Management Center application (formerly known as Sun Enterprise SyMON)?**  
No. SunVTS is installed and run independently.
10. **I know that SunVTS runs on my 32-bit Solaris Operating Environment systems. Will it run on my Solaris 64-bit Operating Environment systems?**

Yes. The `sunvts` command detects which operating environment is running and starts the appropriate SunVTS kernel. You must have the 64-bit SunVTS package installed (`SUNWvtsx`) to run 64-bit tests. See “SunVTS Packages” on page 22.

11. Can I run SunVTS by specifying an option file that is located on another system in the network?

No. The directory where SunVTS expects to find an option file is hardcoded to the local system where SunVTS is running.

12. Are any of the SunVTS tests “data destructive”?

No, SunVTS tests are not data destructive. However, when the `disktest` is run with the `rawtest` write/read mode enabled, it does write data on the selected partition. The test, however, backs up the original data first, and puts it back when the write/read/compare operation is complete. By default, `disktest` runs in read-only mode. In addition, a few intervention mode tests, such as `tapetest` and `disktest` (for the floppy drive) require that you load scratch media that can be written on. Specific test details are explained in the *SunVTS Test Reference Manual*.

13. If I add a device to my system, how do I get SunVTS to recognize the device?

Once the Solaris Operating Environment recognizes the device (either through a `boot -r` or another configuration command), SunVTS uses probe routines to recognize the device. The probe routines run automatically when SunVTS is started. Also, you can have SunVTS probe your system for devices at any time with the SunVTS `reprobe` command.

14. Can I customize any of the SunVTS options or menus?

No.

15. Can I add my own tests to the SunVTS environment?

Yes. See “Adding a Custom Test” on page 214.

16. Why does it take SunVTS a long time to start, and a long time to complete a reprobe?

When SunVTS starts (performing the initial probe), and during a reprobe, SunVTS has to determine your system configuration and initiate the tests that correspond specifically to your system. If your system has many devices, the probe takes a while to complete.

17. One of my devices does not appear in the SunVTS UI. Why not?

First, the Solaris operating environment must recognize a device before SunVTS can see it. Make sure that the operating environment is aware of the device (perform a `boot -r` if necessary). Second, the device might have had a failure when SunVTS probed it. Check the window where you started SunVTS to see if

any errors are reported that might direct you to the problem. Third, while most Sun devices have a corresponding test in SunVTS (and therefore show up in the SunVTS UI), you might have a device that is not supported by SunVTS.

**18. What should I do if the SunVTS appears hung?**

First, give SunVTS some time to run the tests. Sometimes the tests use so much of the system resources that there is a delay in updating the SunVTS UI, and it appears hung. Second, if you determine that SunVTS is indeed hung, it might be caused by one specific test. Try deselecting the test that is indicated as currently running. You can start another UI (see “Starting the SunVTS Kernel and Interface Separately” on page 37) to control SunVTS.

**19. I cannot select trace mode for one of my devices because it does not show up in the trace mode dialog box. Why?**

You are probably using the physical view, which causes some devices to be listed several levels below what the trace dialog box can display. Try selecting the logical view. You will probably be able to locate the device.

**20. How can I identify the name of the test for a given device?**

The test name is enclosed in parentheses and displayed next to the device in the SunVTS UI. For example, the test for the memory device is displayed as `mem(pmemtest)`, where `mem` is the device and `pmemtest` is the name of the test.

**21. I get errors when I run my frame buffer test, but the frame buffer is functional. Why?**

Testing frame buffers requires special considerations. In some cases you must disable the screen saver, enable a frame buffer lock option, and other steps. Sometimes the smallest mouse or keyboard input will cause the test to fail. Refer to the specific frame buffer test in the *SunVTS Test Reference Manual* for details.

**22. I have a script that relies on the old SunVTS message syntax (pre-SunVTS 4.0). The script does not work with the new message syntax. What can I do?**

As of SunVTS 5.0, the `VTS_OLD_MSG` variable is no longer supported. It was used to display test messages in a pre-SunVTS 4.0 format, usually because a script relies on the older format. Now your scripts must accept the current SunVTS message format. You must update your script to use the message format that has had the same format since the SunVTS 4.0 release (see TABLE 4-2).

**23. During a SunVTS installation, I was asked if I wanted to create a new instance of this package. What does this mean?**

This is an indication that one or more of the SunVTS packages are already installed. You should remove them before installing the new ones. See “Installing and Removing SunVTS” on page 21 for the procedures.

24. Sometimes the SunVTS CDE UI (vtsui) fails to start, or while running, it exits with "X Error of failed request". What can I do to correct this problem?

This type of error usually indicates that there is an X server resource issue. Try closing some of your other GUI application windows and restart the SunVTS UI (vtsui).

25. Why do I get the following error when I select clt0d0 in a multi-path configuration?

```
ERROR:clt0d0's alternate path (c2t0d0) is already selected
```

This type of error indicates that a disk drive was already selected and you tried to select it again using one of its alternate multi-path device names. SunVTS does not permit the selection of multiple paths of the same device.





# Creating and Editing Generic Option Files for Advanced Users

---

This appendix provides detailed information about the generic option file. This information is provided to enable users to create or edit a generic option file.

To create a custom generic option file, it is recommended to start with one that is created from a GUI such as CDE. The generic option file is in XML format, and can be edited with any text editor. The following example provides a sample generic option file that uses comments to describe the structure of generic option files:

---

**Note** – Use XML comments when you edit a generic option file. All XML comments should strictly start with `<!--` and should strictly end with `-->`. XML syntax does not allow `---` as a legal comment terminator (because of 3 dashes). A space followed by `-->` is required to terminate XML comments.

---

The following is an example of a host-independent generic option file:

```
<?xml version="1.0" ?>
<!-- This is SunVTS Generic Option file in xml format.
      To edit this file, refer to SunVTS Userguide Manual. -->

<vts_options>
  <vts_info   fmt_version="1.0"   vts_version="5.1"
             src_hostname="myhostname"
             description="First Generated on Thu Apr 11 02:08:57 2002" />

<!-- assert action is used to abort loading if the current
      mode is not an expected mode e.g. offline, etc.
      The assert action used only for cumulative mode load. -->

<action name="assert" args="mapping logical" />
<action name="assert" args="tmode offline" />
```

```

<system_options>
  <option name="system_override" value="enable" />
  <option name="system_concurrency" value="0" />
  .....
  .....
</system_options>

<!-- Example Test: nettest -->
<test name="nettest" >
  <args_format>
    <arg name="target" label="Target Host" op_type="TEXT"
      default="" size="48" />
    <arg name="test" label="Test_Type" op_type="INC_CHOICE"
      default="Random+Increment+Pattern"
      choices="Random,Increment,Pattern" />
    <arg name="timeout" label="Receive_Timeout"
      op_type="NUMERIC"
      default="120" min="1" max="600" />
    <arg name="warn" label="Print_Warning" op_type="TOGGLE"
      default="Disable" choices="Enable,Disable" />
  </args_format>

  <std_options>
    <option name="status" value="enabled" />
    <option name="max_passes" value="0" />
    <option name="max_errors" value="1" />
    <option name="max_time" value="0" />
    <option name="number_of_instances" value="2" />
    <option name="processor_affinity" value="unbound" />
  </std_options>

  <!-- The instance tag can be omitted and test_args tag
  can appear directly inside test tag if options are same
  for all instances. -->

  <instance number="0">
    <!-- Specify arg by name or label. Both are legal. -->
    <test_args>
      <arg name="target" value="" />
      <arg name="test" value="Random+Increment+Pattern"/>
      <arg name="timeout" value="60" />
      <arg name="warn" value="Enable"/>
    </test_args>
  </instance>

  <instance number="1">
    .....
  </instance>

```

```
</test>
</vts_options>
```

The following is an example of a simple generic option file. When this file is loaded, only one hardware test option is cumulatively configured.

```
<?xml version="1.0" ?>
<vts_options>
  <action name="assert" args="tmode offline" />
  <test name="nettest" >
    <test_args>
      <arg name="timeout" value="75" />
    </test_args>
  </test>
</vts_options>
```

---

## C.1 Main Components Overview

This section provides a brief overview of the main components of a the generic options file. The following list specifies the main classes and tags used for the generic options file specification components:

- Actions Specifications — Use `<action ... />` tag
- System Level Options — Use `<system_options> ... </system_options>` tags
- Test Level Options – Use `<test ...> ... </test>` tags
- Group Level Options – Use `<group ...> ... </group>` tags
- Device Level Options – Use `<device ...> ... </device>` tags
- Instance Level Options – Use `<instance ...> ... </instance>` tags
- Standard Options – Use `<std_options> ... </std_options>` tags
- Option Specifications – Use `<option .... />` tag
- Test Program Arguments Options – Use `<test_arg> <arg ... /> ... </test_arg>` tags

## C.1.1 Top Level Components

Out of the preceding list of components, the actions specifications, system level options, test level options, and group level options could appear at the top level immediately inside the `<vts_option>` tags. These top level tags could appear in any order within the `<vts_option>` tags.

## C.1.2 Group Level Options

The group refers to either a logical group or a physical group (depending on the current display mapping) under which tests devices are displayed in UI. Group level options could contain standard options, test level options, and another group level options. Note that nested groups appear in physical mapping display.

## C.1.3 Instance Level Options

Instance level options should appear inside test level or device level options. If the same options should be applied to all instances, the instance tags should not be specified.

## C.1.4 Device Level Options

Device level options, if present, should always appear inside the test level option. If the options to be applied should be the same for all devices for a specific test, device tags should not be specified.

## C.1.5 Test Level Options

Test level options can contain standard options, device level options, instance level options, and test program arguments options. If test program arguments options appear immediately inside the `<test>` tag, then those options are applied for all devices for that test and to all instances for these devices.

## C.1.6 The Standard Options

The standard options tag `<std_options>` could appear inside test level, device level, or group level option tags. The only tag that can appear inside the standard options tag is the `<option .. />` tag. Depending on where the standard options appear (inside test, group, device, or instance), the legal values for `<option ... />` use will be restricted. The legal values for `<option ... />` are described in the subsequent sections.

## C.1.7 Option Specification

The option specification `<option ... />` tag should appear inside `<std_options>` or inside `<system_options>`. When it appears inside `<system_options>`, only the options described in the table for system options are allowed. When it appears inside `<std_options>` the corresponding tables describe the legal options allowed depending on the context.

## C.1.8 Option Descriptions for Standard Options

The legal values for the option tag `<option ... />`, which appears inside `<std_options>` are different depending on the context:

- For group level standard options: See TABLE C-3 on page C-10.
- For test and device level standard options: See TABLE C-4 on page C-12
- For instance level standard options: See Section C.1.15 “Instance Level Standard Options” on page C13.

## C.1.9 Test Program Arguments Specification

The `<test_args>` could either appear immediately inside the `<test>`, `<device>`, or `<instance>` tags.

The hardware test program arguments could be specified in the following format:

```
<test_args>
  <arg name=arg_name value=value />
  <arg name=arg_name value=value />
  <arg label=label value=value />
  <arg ... />
  ...
</test_args>
```

To specify the program argument option, only one attribute of name or label needs to be specified. The `arg` option value is either specified by name or by the corresponding label for that option. The label is the one which is displayed for the corresponding test option in the UI.

## C.1.10 File Header

The first line should be:

```
<?xml version="1.0" ?>
```

The preceding line is required for all XML format files. The rest of the option file should be enclosed within `<vts_options>` and `</vts_options>` tags. Immediately after the start `<vts_options>`, the `<vts_info>` tag is optional. If specified, this tag will be used to check against the file format version that is provided through `fmt_version` attribute. Currently only version 1.0 generic option file format is supported. The other attributes of `<vts_info>` exist only for informational purposes.

## C.1.11 Actions Specification

The format of actions specification is: `<action name="action_name" args="action_args">`. The following table lists the supported `action_name` and `action_args`:

**TABLE C-1** Supported `action_name` and `action_args`

Action Name	Action Args	Description
set	"mapping logical"	Switch to logical mapping display.
set	"mapping physical"	Switch to physical mapping display.
set	"tmode offline"	Switch to offline or functional test mode.
set	"tmode online"	Switch to online test mode.
set	"tmode connection"	Switch to connection test mode.
set	"tmode exclusive"	Switch to exclusive test mode.
select	<i>path_name</i>	Path name of test or group to be selected. For example, "/" for path name selects all.
deselect	<i>path_name</i>	Path name of test or group to be deselected. For example, "/" for path name deselects all.

**TABLE C-1** Supported `action_name` and `action_args` (Continued)

Action Name	Action Args	Description
<code>assert</code>	<code>"mapping logical"</code>	If the current display mapping is not logical, the option file loading will abort. Typically useful in a cumulative load.
<code>assert</code>	<code>"mapping physical"</code>	To assert that the current mapping is physical; if not, abort.
<code>assert</code>	<code>"tmode offline"</code>	To assert that the current mode is offline or functional; if not, abort.
<code>assert</code>	<code>"tmode online"</code>	To assert that the current mode is online; if not, abort.
<code>assert</code>	<code>"tmode connection"</code>	To assert that the current mode is connection; if not, abort.
<code>assert</code>	<code>"tmode exclusive"</code>	To assert that the current mode is exclusive; if not, abort.

## C.1.12 System Level Options

System level options are specified in the following format:

```
<system_options>
  <option name="option_name" value="option_value" />
  <option name="option_name" value="option_value" />
  ....
</system_options>
```

The following table lists the supported system level option names and values:

**TABLE C-2** Supported System Level Option Names and Values

Category	Option Name	Option Value	Description
Global	<code>intervention</code>	<code>"enable"</code> or <code>"disable"</code>	Either enables or disables intervention mode.

**TABLE C-2** Supported System Level Option Names and Values (Continued)

Category	Option Name	Option Value	Description
Threshold Options	max_system_passes	0 or positive number	Maximum number of system passes before stopping. 0 for unlimited. The definition of a system pass is when all selected tests have run one time.
	max_system_errors	0 or positive number	Maximum number of system errors before stopping. 0 for unlimited.
	max_system_time	0 or positive number	Maximum number of minutes for testing before stopping. 0 for unlimited.
Notify Options	send_email	"disabled", "now", "on_error", "periodically", or "on_error&periodically"	Description in the same order: Disable sending email; or Send e-mail now; or only on error; or send e-mail with time period set in log_period option; or send email periodically and on error.
	email_address	"e-mail-addr"	E-mail address where the test status messages are sent to. "root" by default.
	log_period	0 or positive number	Specifies time in minutes between sending test status email. Use 0 to disable.
Schedule Options	auto_start	"enable" or "disable"	If enabled, SunVTS testing will start automatically when invoked without the need to press the start button.
	single_pass	"enable" or "disable"	If enabled, runs only one pass of each selected test.
	system_concurrency	0 or positive number	Maximum number of test instances that can run concurrently. 0 means unlimited.
	group_concurrency	0 or positive number	Maximum number of test instances that can run concurrently in the same group. 0 means unlimited.



**TABLE C-2** Supported System Level Option Names and Values *(Continued)*

Category	Option Name	Option Value	Description
Advanced Options	system_override	"enable" or "disable"	If enabled, the system level options supersede the group-specific, test-specific, and device-specific option settings in favor of the options set at the system level. By default, it is enabled.
	group_override	"enable" or "disable"	If enabled, group level options supersede the test level and device level options in favor of the group options. The system override option supersedes this option. By default, it is enabled.
	group_lock	"enable" or "disable"	If enabled, protects specific group options from being changed from the options set at the system level. The System Override option supersedes this option. By default, it is disabled.
	test_lock	"enable" or "disable"	If enabled, protects specific device level options from being changed from the options set at the test level, group level, or system level. The System Override and Group Override options supersede this option. By default, it is disabled.
Test Execution Options	See Description	See Description.	The test execution options could appear in system level, group level, test level, or device level. See TABLE C-5 for a description of the test execution standard options.

## C.1.13 Group Level Standard Options

Group level standard options are specified in following format:

```
<group>
  <std_options>
    <option name="option_name" value="option_value" />
    <option name="option_name" value="option_value" />
    ....
  </std_options>
  .....
</group>
```

The following table provides a list of supported group level standard options:

**TABLE C-3** Supported Group Level Standard Options

Category	Option Name	Option Value	Description
General	status	"enable" or "disable"	Either enables or disables all the devices in the group
Schedule Options	group_concurrency	0 or positive number	Maximum number of test instances that can run concurrently in this group. 0 means unlimited.

**TABLE C-3** Supported Group Level Standard Options (*Continued*)

Category	Option Name	Option Value	Description
Advanced Options	group_override	"enable" or "disable"	If enabled, group level options supersede the test level and device level options in favor of the group options. The system override option at the system level supersedes this option. By default, it is enabled.
	group_lock	"enable" or "disable"	If enabled, protects specific group options from being changed from the options set at the system level (the system override option at the system level supersedes this option.) By default, it is disabled.
	test_lock	"enable" or "disable"	If enabled, protects specific device level options from being changed from the options set at the test level, group level, or system level. (the system override option at the system level and group override options supersede this option.) By default, it is disabled.
Test Execution Options	See description	See description	The test execution options could appear in either the system level, group level, test level, or device level. See TABLE C-5 for a description of the test execution standard options.

## C.1.14 Test and Device Level Standard Options

The standard options in the test level are specified in the following format:

```

<test name=test_name >
  <std_options>
    <option name="option_name" value="option_value" />
    <option name="option_name" value="option_value" />
    ....
  </std_options>
  .....
</test>

```

The standard options in the device level are specified in the following format:

```

<test name=test_name >
  <device name=device_name >
    <std_options>
      <option name="option_name" value="option_value" />
      <option name="option_name" value="option_value" />
      ....
    </std_options>
  </device>
  .....
</test>

```

The legal standard options for the test level and device level are the same. The test level standard options are applied to all devices being tested by that test. The device level standard options are applied to only the specified device.

**TABLE C-4** Test and Device Level Standard Options

Category	Option Name	Option Value	Description
General	status	"enable" or "disable"	When specified at the test level, enables/disables testing for all devices for that test. When specified at the device level, enables/disables testing on the specified device.
Advanced Options	test_lock	"enable" or "disable"	If enabled, protects specific device level options from being changed from the options set at the test level, group level, or system level. The system override at the system level and group override options supersede this option. By default, it is disabled.
Test Execution Options	See Description	See Description	The test execution options could appear in system level, group level, test level, or device level. See TABLE C-5 for a description of the test execution standard options.

## C.1.15 Instance Level Standard Options

The instance level standard options are specified in following format:

```
<test name=test_name >
  .....
  <instance>
    <std_options>
      <option name="option_name" value="option_value" />
    </std_options>
    ....
  </instance>
  .....
</test>
```

The only instance level standard option allowed is `processor_affinity`. For the description of `processor_affinity` option, refer to TABLE C-5.

## C.1.16 Test Execution Standard Options

The test execution options could appear at the system level, group level, test level, or device level. Depending on the different override options enabled or disabled, test execution options at certain levels override the options specified in another level. The following table describes the test execution valid option names and values.

TABLE C-5 Test Execution Standard Options

Option Name	Option Value	Description
<code>verbose</code>	"enable" or "disable"	Displays verbose messages in the SunVTS messages window.
<code>core_file</code>	"enable" or "disable"	If enabled, creates a core file when a test session results in a core dump.
<code>run_on_error</code>	"enable" or "disable"	If enabled, SunVTS continues to run until the max errors value is reached. If disabled, irrespective of max errors, SunVTS stops testing on encountering first test failure. When <code>run_on_error</code> is enabled, the tests are also invoked with a run on error flag.
<code>max_passes</code>	non-negative-number	Number of test passes before stopping. Default 0 means unlimited.

**TABLE C-5** Test Execution Standard Options *(Continued)*

Option Name	Option Value	Description
<code>max_errs</code>	non-negative-number	Number of test errors before stopping. Default 0 means unlimited.
<code>max_time</code>	non-negative-number	Time limit in minutes before stopping. 0 means there is no time limit.
<code>max_delay</code>	non-negative-number	Specifies, in seconds, the amount of time to pause between execution of a test. Valid range is between 0 to 60. By default, 0 performs no delay.
<code>number_of_instances</code>	non-negative-number	Specifies number of instances to run each scalable test.
<code>processor_affinity</code>	unbound, sequential, or processor-id	Specifies how the instances of the test should bind to different processors. Applicable only in multi-processor systems. <code>unbound</code> is the default and does not bind the test instances to any processor. <code>sequential</code> binds all the test instances to different processors in the system as the test processes are created. Even if the number of processors is higher than instances, the sequential binding will eventually bind test instances to all the processors one after another as the test passes and progresses. Specifying a <code>processor-id</code> (non-negative-number) binds the test instance(s) to a specific processor.

# SunVTS Global Configuration File

## `sunvts.conf`

---

This appendix describes the SunVTS 5.1 `sunvts.conf` configuration file. As of SunVTS 5.1, an optional `sunvts.conf` configuration file is supported. The path name of this configuration file is `/etc/opt/SUNWvts/sunvts.conf`.

---

**Note** – This file is optional; therefore, it is not required to use SunVTS.

---

The configuration file `/etc/opt/SUNWvts/sunvts.conf` is read by the kernel prior to the system probing the devices.

A reference copy of the configuration file, `sunvts.conf.example` is provided in the `/etc/opt/SUNWvts` directory. This file is created during installation with comments describing the supported parameters. You can modify the example file or create a new one and save it as `sunvts.conf` to be read by the kernel.

The following are the currently supported configuration file parameters.

**TABLE D-1** Supported `sunvts.conf` File Parameters.

---

Action Name	Action Args
<code>DEF_SCALABLE_INST</code>	The user can specify the default number of scalable instances.
<code>DISABLE_PROBE</code>	List of test names for which probe should be disabled. By default, nothing is disabled.
<code>ENABLE_PROBE</code>	List of test names for which probe should be enabled. By default, all the tests are enabled.
<code>MAX_LOG_SIZE</code>	Maximum size of each log file in MB can be specified using this parameter. By default, the maximum log size is set to 5 MB.

---

---

**Note** – Normally, either `ENABLE_PROBE` or `DISABLE_PROBE` is specified; when both `ENABLE_PROBE` and `DISABLE_PROBE` options are specified, probe is performed only on those tests that are listed under `ENABLE_PROBE` but not under `DISABLE_PROBE`.

---

Lines starting with a '#' are considered comments. Each line specifies a parameter and has the following format:

```
PARAMETER VALUE1 VALUE2 ... VALUEn
```

The following is an example `/etc/opt/SUNWvts/sunvts.conf` file:

```
#
# This is system config file for SunVTS 5.1
# Path name: /etc/opt/SUNWvts/sunvts.conf
#
# Syntax for the entries :
# PARAMETER  VALUE1 VALUE2..... VALUEn
# Examples:
#
DEF_SCALABLE_INST 4
#
# DISABLE_PROBE  cputest mptest
#
ENABLE_PROBE     disktest envtest fptest
#
MAX_LOG_SIZE    3
```

The preceding sample `sunvts.conf` file specifies the maximum number of scalable instances as 4. This also specifies that the probe should be performed only for `disktest`, `envtest`, and `fptest`. Note, that `DISABLE_PROBE` has been commented out. The maximum log size is set to 3 MB.



# Index

---

## A

- accessing SunVTS man pages, 2-5
- Across All Instances, A-20
- Across All Instances (CDE), 4-13
- Across\_All\_Instances (TTY), 5-12
- additional features with the
  - CDE UI, 4-18
  - TTY UI, 5-17
- Applications and SunVTS, 3-1
- Apply, 4-12
- Apply to All, 4-13
- Apply to Group, 4-13
- arrow keys, 5-4
- asterisk, 4-14
- Auto Config mode, 1-5, 4-5, 5-6
- Auto Start, A-16
- Automatic Configuration, 4-18, A-6
  - Comprehensive testing, A-9
  - Confidence testing, A-9
    - dialog box, 4-19, A-9
    - using the, 4-19
- auto-start feature (CDE), 4-28, 5-25

## B

- background mode, 1-7
- backspace key, 5-4
- backup logs, 4-21, 5-19
- basic security, 2-7
- binding a test (CDE), 4-25
- binding a test (TTY), 5-22
- boot -r, B-3
- BYPASS\_FS\_PROBE, 2-12

## C

- CDE display mode, 4-6
- CDE main window, 4-2
- CDE UI
  - Status panel, 4-4, A-5
  - Test mode panel, 4-5, A-6
- CD-ROM drive preparation, 3-4
- changing windows (TTY), 5-4
- checking for test messages (TTY), 5-14
- checkmark, 4-9
- clearing the current results (CDE), 4-17
- closing the Log file window (CDE), 4-15
- collapsing the list of devices (CDE), 4-8
- color indications, 4-14
- Commands menu (DS), A-11
- communication port preparation, 3-4
- complete service name, SEAM, 2-4, 2-10
- Comprehensive testing, A-9
- Confidence testing, A-9
- configuring a test session (CDE), 4-4
- configuring a test session (TTY), 5-4
- configuring tasks, 5-29
- Connect to Host field, 4-20, A-10
- connecting to another host (CDE), 4-20
- connecting to another host (TTY), 5-18
- Connection test mode, 1-5, 4-5, A-6
- Connection test mode (TTY), 5-6
- Consistent testing, 4-18
- Control panel (TTY), 5-3, A-7
- controlling security, 2-10
- Core File option, A-17
- core file option, 4-26
- core file option (TTY), 5-23

- creating DSched sequences (TTY), 5-31
- creating DSched tasks (TTY), 5-29
- Cumulative errors (CDE), 4-4, A-5
- current device under test, 4-14
- Current Loop (DS), A-12
- custom test, installing a, 2-14

## D

- data destructive question, B-3
- debugging features (CDE), 4-25
- debugging features (TTY), 5-22
- decreasing testing intensity (CDE), 4-24
- default directory, B-2
- default installation directory (/opt), 2-3
- default security, 2-11
- Default selection, 4-9
- Defining DSched tasks, 4-32
- Defining the sequence loops, 4-36
- Defining the Task Sequence, 4-34
- Delete button (DS), A-12
- deleting a DSched sequence (TTY), 5-35
- deleting a log (CDE), 4-15
- deleting DSched tasks (TTY), 5-31
- deleting text in field (TTY), 5-4
- description of SunVTS, 1-1
- Deterministic Scheduler, 4-30, 5-27
- device, 4-9
- device group, 4-9
- device groups (TTY), 5-8
- device preparation, 3-4
- Device Selection panel, 4-9
- device-level options (CDE), 4-12
- device-level options (TTY), 5-11
- devices in the System Map (CDE), 4-8
- devices not appearing question, B-3
- dialog boxes, A-1
- disabling tracing (CDE), 4-22
- discarding pop-up menus (TTY), 5-4
- diskette drive preparation, 3-4
- display mode (CDE), 4-6
- display mode (TTY), 5-7
- displaying a menu (TTY), 5-4
- ds\_idle, 4-4, A-5
- ds\_running, 4-4, A-5
- ds\_suspended, 4-4, A-5
- dsched, 5-28

## E

- Edit button (DS), A-12
- Edit panel (DS), A-12
- Elapsed test time in CDE Status panel, 4-4, A-5
- Email Address field, A-13
- email notification (CDE), 4-21
- environmental variables, 2-12
- ERROR, 4-16
- error log, 4-21, 5-19
- error messages (CDE), 4-15
- error status log (CDE), 4-14, 5-14
- escape key, 5-4
- expanding the list of devices (CDE), 4-8

## F

- FATAL, 4-16
- frequently asked questions, B-1
- Functional test mode, 1-5
- Functional test mode (CDE), 4-5, A-6
- Functional test mode (TTY), 5-6

## G

- getting started
  - SunVTS CDE UI, 4-1
  - SunVTS TTY UI, 5-1
- graphic test considerations, 3-2
- Group Concurrency, A-16
- Group Concurrency (CDE), 4-24
- Group Concurrency (TTY), 5-21
- Group Lock, A-8
- Group Override, A-8
- grouping menu (TTY), 5-7
- group-level options, 4-12
- group-level options (TTY), 5-11

## H

- Hostname in Status panel, 4-4, A-5
- Hostname List, 4-20, A-10

## I

- Idle, 4-4, A-5

- increase testing intensity (CDE), 4-24
- INFO, 4-15
- information log, 4-21, 5-19
- information log (CDE), 4-15, 5-14
- Insert button (DS), A-12
- inserting DSched tasks (TTY), 5-34
- installation requirements, 2-3
- Installing SunVTS, 2-3
- instance, test, 4-24
- internationalization, 2-12
- interprocess communication protocols, 1-7
- Intervention (CDE), 4-9
- Intervention (TTY), 5-7
- IPC (interprocess communication), 1-7

## K

- Kerberos security, 2-7
- kernel, SunVTS, 1-7
- keyboard commands (TTY), 5-4

## L

- LANG variable, 2-13
- Latest SunVTS news, 3-2
- List... (TTY), 5-30
- Load, DSched (TTY), 5-30
- localized environment, 2-12
- localized fonts, using, 2-13
- locks (CDE), 4-23
- locks (TTY), 5-19
- Log button (CDE), 4-15
- log file sizes (CDE), 4-21, 5-19
- Log Period, A-13
- log\_files menu (TTY), 5-15
- logical view
  - system mapping (CDE), 4-6
  - system mapping (TTY), 5-7
- logs (CDE), 4-14
- logs (TTY), 5-14
- loops, 4-31, 5-28

## M

- main windows, A-1
- man pages, 2-2, 2-5

- MANPATH, 2-12
- Max Error option, A-17
- Max Passes option, A-17
- Max System Errors, A-21
- Max System Log Size, 4-22, A-21
- Max System Passes, A-21
- Max System Passes, DSched, 5-29
- Max System Time, A-21
- Max Time option, A-18
- Menu Bar (CDE), A-3
- Message console (TTY), 5-3
- Message panel (CDE), 4-16
- Message panel (TTY), A-7
- message syntax, 4-17
- message syntax (old version), B-4
- message type (CDE), 4-16
- message types (TTY), 5-16
- Model in CDE Status panel, 4-4, A-5
- modifying DSched tasks (TTY), 5-30
- monitoring a test session (CDE), 4-13
- Monitoring a test session (TTY), 5-13
- moving between options (TTY), 5-4
- moving to another window (TTY), 5-4
- multiple copies of a test, 4-24
- multiple versions, B-2
- multiprocessor systems and scaling, 4-24

## N

- non-english environments, 2-12
- notification by email (CDE), 4-21
- notification by email (TTY), 5-18
- Number of Instances option, A-18

## O

- old message syntax, B-4
- Open System Map command (CDE), 4-8
- operating environment, B-3
- Option File List, A-14, A-15
- option file support question, B-3
- Option Files (CDE), 4-11, 4-26
- Option Files (TTY), 5-23
- Option Files field, A-14, A-15
- options
  - device-level (CDE), 4-10
  - system-level (CDE), 4-10

overriding test options (CDE), 4-23  
overriding test options (TTY), 5-20

## P

packages, 2-2  
Performance meter (CDE), A-4  
physical view

- system mapping (CDE), 4-6, 5-7

`pkgadd` installation, 2-3  
`pkginfo`, 2-3  
predefined test options, A-9  
predetermined test options, 4-18  
preparation of devices, 3-4  
preserving test options (CDE), 4-23  
preserving test options (TTY), 5-19  
principal, SEAM, 2-4, 2-10  
Print the log, 4-15  
probable causes, 4-16  
probing question, B-3  
procedures overview, 3-3  
processor affinity, 4-25  
Processor Affinity option, A-20  
protocols, interprocess communication, 1-7

## Q

Quick Access buttons (CDE), A-4  
Quit UI Only (DS), A-11  
quitting DSched (TTY), 5-36  
quitting

- TTY user interface, 5-4

quitting SunVTS, 3-13

## R

recommended corrective actions, 4-16  
Record and Replay (CDE), 4-29  
Record and Replay (TTY), 5-26  
refreshing a TTY window, 5-4  
Reliable testing, 4-18  
remote machine

- displaying devices, 3-15
- using SunVTS on a, 3-9

remote system (CDE), 4-20  
Remove task (DSched (TTY), 5-31

removing an option file, 4-28  
removing DSched tasks (TTY), 5-33  
Removing SunVTS, 2-16  
reproducing a sequence of events, 4-29  
requirements for starting SunVTS, 3-2  
resetting DSched (TTY), 5-36  
resetting results (CDE), 4-14  
Resetting test session results (TTY), 5-17  
resuming a test session (CDE), 4-29  
resuming a test session (TTY), 5-26  
return key, 5-4  
reviewing results (CDE), 4-14  
Run button (DS), A-12  
Run On Error option, A-17  
Run panel (DS), A-12  
running a test session (CDE), 4-13  
running a test session (TTY), 5-13  
running DSched (TTY), 5-35  
running multi-user mode, B-2  
running through a serial terminal (TTY), 5-2  
Runtime Considerations, 3-1

## S

saving a configuration (CDE), 4-26  
saving a configuration (TTY), 5-23  
scaling a test session (CDE), 4-24  
scaling a test session (TTY), 5-21  
Schedule Option menu (TTY), 5-21  
Schedule options (CDE), 4-13  
scripts and SunVTS, 4-16  
scrolling (TTY), 5-4  
SEAM security (Kerberos), 2-7  
security, 1-5, 2-7

- basic, 2-7
- controlling, 2-10
- default, 2-11
- installing, 2-4
- SEAM, 2-7
- switching, 2-11

Select Devices panel (CDE), A-5  
Select mode panel (CDE), A-6  
Select Test panel, A-22  
selecting a test mode (CDE), 4-5, A-6  
selecting and applying a menu choice (TTY), 5-4  
selecting devices (CDE), 4-9  
selecting devices (TTY), 5-8  
Send Email menu, A-13

- sequence, 4-31, 5-28
- Sequence Name (DS), A-11
- Sequence Name field, DSched (TTY), 5-31
- service name, SEAM, 2-4, 2-10
- Set Task Insert Mark, 5-34
- set\_options menu (TTY), 5-10
- setting options values (TTY), 5-10
- shifting to another window (TTY), 5-4
- Show DS, 5-29
- simplified configuration, 4-18
- simultaneous testing, 4-24
- Single Pass, A-16
- Solaris Supplement CD, B-2
- spacebar, 5-4
- Start DS, 5-29
- starting a test session (CDE), 4-13
- starting a test session (TTY), 5-13
- starting DSched (TTY), 5-28
- starting the CDE interface (vtsui), 3-8
- starting the Deterministic Scheduler, 4-31
- starting the TTY interface (vtstty), 3-9
- Status, 4-12
- Status panel (CDE), 4-4, A-5
- Status panel (TTY), 5-3, 5-4, A-7
- Stop button (CDE), 4-14
- stopping a test session (CDE), 4-14
- stopping a test session (TTY), 5-14
- Stress option, A-17
- Sun Management Center, B-2
- SunVTS
  - architecture, 1-6
  - description of, 1-1
  - dialog boxes, A-1
  - frequently asked questions, B-1
  - kernel, 1-7, 3-7
  - main windows, A-1
  - messages (CDE), 4-15
  - procedures, 3-3
  - quitting, 3-13
  - scripts, and, 4-16
  - starting, 3-5
  - TTY UI, 5-1
- sunvts command, 3-5
- sunvts\_sec\_gss file, 2-11
- SUNWvts, 2-2
- SUNWvtsmn, 2-2
- SUNWvtsx, 2-2
- Supplement CD, 2-4
- Suspend, 4-4, A-5

- suspending a test session (CDE), 4-29
- suspending a test session (TTY), 5-26
- swap space considerations, 3-2
- switching security, 2-11
- system call, 4-22
- System Concurrency, A-16
- System Concurrency (CDE), 4-24
- System Concurrency (TTY), 5-21
- System load and SunVTS, 3-1
- System Map (CDE), A-6
- System Map display mode (CDE), 4-6
- System Map, devices in (CDE), 4-8
- system mapping
  - logical view (CDE), 4-6
  - logical view (TTY), 5-7
  - physical view (CDE), 4-6, 5-7
- system message log (CDE), 4-15, 5-15
- System Override, A-8
- System passes in CDE Status panel, 4-4, A-5
- system-level test options (CDE), 4-10

## T

- tab key, 5-4
- tape drive preparation, 3-4
- Task File (TTY), 5-30
- Task Manager menu (TTY), 5-29
- tasks, 4-31, 5-28
- Tasks in Sequence list (DS), A-11
- telnet and SunVTS, 3-11
- terminating the kernel, 3-13
- terminating the user interface, 3-13
- Test
  - preparation, 3-4
- Test Advanced (CDE), 4-13
- Test categories, 1-2
- test concurrency, 4-24
- Test Execution, 4-11, 4-24
- Test Execution (CDE), 4-13
- test instance, 4-25
- test instance (CDE), 4-24
- test instance (TTY), 5-21
- Test Lock, A-8
- Test Message panel (CDE), 4-14, A-6
- test modes, 1-5
  - Connection, 1-5
  - Functional, 1-5
- test modes (CDE), 4-5

- Test modes (TTY), 5-6
- test name question, B-4
- test option (TTY), 5-9
- test options (CDE), 4-10
- Test Parameter Options, 4-12
- Test\_Groups panel (TTY), 5-3, 5-8, 5-11, A-7
- test\_mode (TTY), 5-6
- Test\_Options menu (TTY), 5-11
- test\_select (TTY), 5-8
- testing duration (CDE), 4-13
- testing duration (TTY), 5-13
- Testing status in Status panel, 4-4, A-5
- Thresholds, 4-11, 4-21
- tip and SunVTS, 3-11
- Total Loop (TTY), 5-36
- Total Loops (DS), A-12
- trace a test (CDE), 4-22
- trace mode question, B-4
- truss, 4-22
- TTY main window, 5-3
- TTY UI, 5-1

## U

- user interfaces, description of, 1-3
- using the SunVTS CDE UI, 4-1
- using the SunVTS TTY UI, 5-1

## V

- variables, SunVTS, 2-12
- VERBOSE, 4-15
- verbose mode (CDE), 4-14
- verbose mode (TTY), 5-23
- Verbose option, A-17
- viewing devices (CDE), 4-8
- viewing devices (TTY), 5-9
- viewing DSched sequences (TTY), 5-33
- viewing system calls, 4-22
- VTS\_CMD\_HOST, 2-12
- VTS\_OLD\_MSG, 2-12
- VTS\_PM\_PATH, 2-12
- vtsk command, 4-20
- vtspoke command, 3-13
- vtstty, 3-9
- vtsui, 3-8

## W

- WARNING, 4-16
- Within Instance, A-20
- Within Instance (CDE), 4-12
- Within\_Instance (TTY), 5-12

## X

- x86 systems, B-2