

Solstice DiskSuite Tool 4.0 User's Guide

2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

Part No: 802-1724-10
Revision A, March 1995



© 1995 Sun Microsystems, Inc.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX® and Berkeley 4.3 BSD systems, licensed from UNIX System Laboratories, Inc., a wholly owned subsidiary of Novell, Inc., and the University of California, respectively. Third-party font software in this product is protected by copyright and licensed from Sun's font suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, the Sun logo, Sun Microsystems, Sun Microsystems Computer Corporation, SunSoft, the SunSoft logo, Solaris, SunOS, Solstice, OpenWindows, DeskSet, Online: DiskSuite, ONC, ONC+, and NFS are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and certain other countries. UNIX is a registered trademark of Novell, Inc., in the United States and other countries; X/Open Company, Ltd., is the exclusive licensor of such trademark. OPEN LOOK® is a registered trademark of Novell, Inc. PostScript and Display PostScript are trademarks of Adobe Systems, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCstorage, SPARCware, SPARCcenter, SPARCclassic, SPARCcluster, SPARCdesign, SPARC811, SPARCprinter, UltraSPARC, microSPARC, SPARCworks, and SPARCcompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and Sun™ Graphical User Interfaces were 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.

X Window System is a product of the Massachusetts Institute of Technology.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.



Contents

1. Introduction	1
DiskSuite and DiskSuite Tool Functionality	2
Getting Help	3
Supported Software and Hardware	3
Disk Naming	4
Definition of Terms	4
2. Installation and Setup	9
Preparing for Installation	10
Local Installation	10
Remote Installation	11
Adding Packages	12
Viewing Installed Packages	21
Removing Packages	22
Path Requirements	23
Viewing AnswerBook Online Documentation	24

What to Do Next	24
3. Overview of DiskSuite Tool	25
DiskSuite Description	25
Metadevices	26
Concatenations and Stripes	27
Mirroring	27
UFS Logging.	27
Hot Spares	28
Disksets.	28
RAID Devices.	28
Metadevice State Database Replicas	28
Expanding Mounted File Systems.	29
DiskSuite Tool's User Interface	29
Starting DiskSuite Tool	30
Metadevice Editor Window	32
Disk View Window	38
Information Windows.	42
Browsers	46
Dialog Boxes.	52
Configuration Log Window	53
Problem List Window	54
Accessing and Using Help	55
DiskSuite Recovery and Guidelines	56
Recovery From Failed Boots.	56

Configuration Guidelines	57
4. Concatenating and Striping	59
Overview of Concatenating and Striping	59
Using Concatenations and Stripes	61
Hardware and Software Considerations	63
Defining Metadevice Configurations	63
Concatenated Metadevices	63
Striped Metadevices	68
Metadevices Defined as Concatenated Stripes	74
Replacing Failed Components	80
Clearing Concatenations and Stripes	84
Concat Information Window	87
Stripe Information Window	89
Checking the Status of Concatenations and Stripes	91
5. Mirroring	93
Overview of Mirroring	94
Operation of Mirrors	94
Creating Mirrors	95
Mirroring Unused Slices	95
Mirroring Existing File Systems	100
Unmirroring File Systems	108
Mirroring File Systems You Cannot Unmount	110
Unmirroring File Systems You Cannot Unmount	120
Recording and Booting From Alternate Root File Systems . . .	120

Reconfiguring Submirrors	121
Replacing Submirrors	121
Attaching and Detaching Submirrors	126
Placing Submirrors Online and Offline.	130
Replacing and Enabling Submirror Components	134
Using Mirrors for Online Backup	137
Mirror Information Window	137
Checking the Status of Mirrors	141
6. UFS Logging	143
Overview of UFS Logging	143
How UFS Logging Works.....	144
Space Requirements for Logs.....	145
Selecting Which File Systems Should Be Logged	145
Determining the Location of Logs.....	146
How to Set Up UFS Logging on an Existing File System	146
How to Share a Log Between File Systems.....	150
Removing UFS Logging	154
Logging a File System That You Cannot Unmount.....	157
Removing UFS Logging From a File Systems You Cannot Unmount.	162
Using Metadevices and Mirrors as Trans Components	162
Trans Information Window	163
Checking the Status of Trans Metadevices	165
Recovering From Device Errors and File System Panics	165

7. Hot Spares	167
Overview of Hot Spares	167
Hot Spare Conditions to Avoid	168
Defining Hot Spares.	168
Associating Hot Spare Pools	172
Manipulating Hot Spare Pools	175
Adding Hot Spares	176
Deleting Hot Spares.	179
Replacing Hot Spares	180
Enabling Hot Spares	183
Changing the Associated Hot Spare Pool	186
Hot Spare Information Window.	188
Checking the Status of Hot Spares.	191
8. Administering Metadevices in Disksets.	193
Overview of Disksets.	193
Administering Disksets.	195
9. RAID Devices	197
RAID Overview	197
Operation of RAID	198
Hardware and Software Considerations.	199
Creating RAID Metadevices.	199
Reconfiguring RAID Metadevices	204
Concatenating Components	204
Assigning a Hot Spare Pool	208

Replacing Components	210
RAID Information Window	214
Checking the Status of RAID Metadevices	217
10. State Database Replicas	219
Overview of the State Database Replicas	219
Performing the Memory Service	220
Planning Locations of Replicas	221
Creating the Initial State Database Replicas	222
Adding Replicas	225
Removing Replicas	228
Metadevice State Database Information Window	231
Checking the Status of Replicas	234
11. Expanding a File System	235
File System Expansion Overview	235
Nonexpandable File Systems	236
Growing File Systems	236
Adding a Slice to a Metadevice	237
Adding a Slice to a RAID Metadevice	242
Adding Slices to Submirrors	247
A. Example Use of DiskSuite Tool	253
Overview of the Example	253
Planning for Availability	256
Planning the Layout of the Data	257
Setting Up the Configuration Using DiskSuite Tool	262

B. DiskSuite Tool Messages.....	265
State Information Terms	265
Metadevice Editor Messages	266
Dialog Box Error Messages.....	267
Dialog Box Warning Messages.....	271
Dialog Box Information Messages.....	279
Metadevice Editor Window Messages	280
Messages Displayed When Dragging an Object	281
Disk View Window Messages	286
Messages Displayed When Pointing at an Object	287
Messages Displayed When Dragging an Object	287
Log Messages	288
Notice Log Messages	288
Warning Log Messages	289
Panic Log Messages.....	292
C. Man Page	295

Figures

Figure 3-1	DiskSuite Tool Metadevice Editor Window	32
Figure 3-2	Button Panel Selections.	34
Figure 3-3	Template Objects	35
Figure 3-4	Device List	36
Figure 3-5	Panner Region	37
Figure 3-6	Disk View Window.....	38
Figure 3-7	Drop Site Objects.....	40
Figure 3-8	Disk View Objects	41
Figure 3-9	Legend Region	41
Figure 3-10	Disk Information Window.....	43
Figure 3-11	Slice Information Window.....	45
Figure 3-12	Slice Browser Window	46
Figure 3-13	Find Window	49
Figure 3-14	Slice Filter Window.....	50
Figure 3-15	Example Dialog Box	52
Figure 3-16	Configuration Log Window.....	53

Figure 3-17	Problem List Window	54
Figure 3-18	DiskSuite Tool Help Utility	55
Figure 4-1	Concatenation of Three 327-Mbyte Drives	60
Figure 4-2	Striping of Three 327-Mbyte Drives	61
Figure 4-3	Concatenated Metadevice	63
Figure 4-4	Striped Metadevice	68
Figure 4-5	Concatenation of Two Stripes Into a Metadevice	74
Figure 4-6	Concat Information Window	87
Figure 4-7	Stripe Information Window	89
Figure 5-1	Mirror Information Window	138
Figure 6-1	Trans Information Window	163
Figure 7-1	Hot Spare Information Window	189
Figure 8-1	Example of a Diskset	194
Figure 9-1	RAID Information Window	214
Figure 10-1	Metadevice State Database Information Window	232
Figure A-1	Hardware View of the Example Configuration	255
Figure A-2	Hierarchy of the Example Configuration	257
Figure A-3	View of the Data Layout	260
Figure A-4	Hardware View of the Layout	261

Tables

Table 3-1	DiskSuite Tool Mouse Model	29
Table 3-2	Metadevice Editor Window Menu Choices.....	33
Table 3-3	Disk View Window Menu Choices.....	39
Table 3-4	Disk Information Screen Functionality	44
Table 3-5	Slice Information Screen Functionality	45
Table 3-6	Slice Browser Device List Information.....	47
Table 3-7	Metadevice Browser Device List Information.....	48
Table 3-8	Hot Spare Pool Device List Information	48
Table 3-9	Slice Filter Window Items	51
Table 3-10	Dialog Boxes Information.....	53
Table 3-11	DiskSuite Tool Help Buttons	56
Table 4-1	Concat Information Window Functionality	88
Table 4-2	Slice Information Window Functionality.....	90
Table 4-3	Concat/Stripe Status Keywords	91
Table 4-4	Stripe Status Keywords	91
Table 5-1	Mirror Information Window Functionality.....	139

Table 5-2	Mirror Status Keywords	142
Table 6-1	Trans Information Window Functionality	164
Table 6-2	Trans Metadevice Status Keywords	165
Table 7-1	Hot Spare Pool Information Window Functionality	190
Table 7-2	Hot Spare Pool Status Keywords	191
Table 9-1	RAID Information Window Functionality	215
Table 9-2	RAID Status Keywords	217
Table 10-1	Metadevice State Database Information Window Functionality	233
Table 10-2	MetaDB Status Keywords	234
Table A-1	Example Disk Partitioning	262

Preface

The *Solstice DiskSuite Tool 4.0 User's Guide* provides all information necessary for using the DiskSuite Tool to create and manipulate Solstice™ DiskSuite™ 4.0 configurations.

The Solstice DiskSuite software package offers a pseudo device driver (called a *metadisk driver*) that provides better performance, greater capacity, and improved availability of data.



Caution – Before using DiskSuite Tool, you must completely understand the components of DiskSuite 4.0 software. Without this knowledge, data can be lost. It is strongly recommended that in addition to reading this manual you read the *Solstice DiskSuite 4.0 Administration Guide*.

Solstice DiskSuite allows for up to three-way mirroring of any file system including root (/), /usr, and swap. Other features of DiskSuite include online concatenation of physical drives, online expansion of file systems, disk striping, hot spares, RAID, creation of disksets, and UFS logging.

DiskSuite Tool runs only on Solaris® 2.4 or a later Solaris 2.x release.

Who Should Use This Book

System administrators and others with the task of administering disk configurations and performing disk maintenance will find this manual to be a valuable resource. Much of the information in this book is targeted towards administrators with experience performing disk maintenance. This manual is not meant to be used by junior administrators.

How This Book Is Organized

This document has 11 chapters and three appendixes.

Chapter 1, “Introduction,” introduces the basic concepts of DiskSuite Tool, lists the peripherals supported, and defines the general terminology used in this document.

Chapter 2, “Installation and Setup,” provides information on what you need to do before using DiskSuite Tool, including installation and initial setup instructions for the software.

Chapter 3, “Overview of DiskSuite Tool,” offers a high-level overview of the functionality included with DiskSuite and the DiskSuite Tool. The various screens you will use when using DiskSuite Tool are shown and explained.

Chapter 4, “Concatenating and Striping,” provides conceptual details and procedures for using DiskSuite Tool to create metadevices consisting of either concatenations, stripes, or concatenated stripes.

Chapter 5, “Mirroring,” includes information on the operation of mirrors, using mirrors to recover from a single-component failure, and the configuration of mirrors.

Chapter 6, “UFS Logging,” provides information on using DiskSuite Tool to set up and use the UNIX® file system logging facility.

Chapter 7, “Hot Spares,” describes how to define and use hot spare pools and includes information on conditions to avoid when using hot spares.

Chapter 8, “Administering Metadevices in Disksets,” gives the procedure for displaying disksets using DiskSuite Tool.

Chapter 9, “RAID Devices,” discusses redundant arrays of inexpensive disk (RAID) device configuration information.

Chapter 10, “State Database Replicas,” provides an overview of the state database and discusses proper use.

Chapter 11, “Expanding a File System,” provides instructions for expanding mounted UNIX file systems.

Appendix A, “Example Use of DiskSuite Tool,” gives a complete example of how to configure a DiskSuite configuration using DiskSuite Tool.

Appendix B, “DiskSuite Tool Messages,” explains the status, error, and log messages displayed by DiskSuite Tool and the underlying DiskSuite software.

Appendix C, “Man Page,” is a printed copy of the man page for the `metatool(1M)` command, which is also included on the installation CD-ROM.

Related Books

You must use the *Solstice DiskSuite 4.0 Administration Guide* with this manual. The *Solstice DiskSuite 4.0 Administration Guide* contains in-depth discussions of all parts of the DiskSuite product in addition to the command-line usage.

Other Sun documentation related to disk maintenance and configuration includes the *Solaris 2.4 File System Administration* manual.

What Typographic Changes Mean

The following table describes the typographic changes used in this book.

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
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. machine_name% You have mail.
AaBbCc123	What you type, contrasted with on-screen computer output	<div>machine_name% su Password:</div>
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be root to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

Table P-2 Shell Prompts

Shell	Prompt
C shell prompt	machine_name%
C shell superuser prompt	machine_name#
Bourne shell and Korn shell prompt	\$
Bourne shell and Korn shell superuser prompt	#

Introduction



This chapter introduces and gives an overview of the Solstice DiskSuite Tool software.

This introductory chapter provides the following:

- A brief description of the DiskSuite Tool
- Information on how to get help
- A list of the peripherals supported by DiskSuite
- A list of DiskSuite and DiskSuite Tool terms and their definitions

The existing DiskSuite software enables you to create and manipulate metadevices using the command-line interface. The advantage of DiskSuite Tool (`metatool(1M)`) is the graphical representation of all metadevices and the convenient drag-and-drop interface.

All the functionality available using the DiskSuite command-line interface is available using the DiskSuite Tool, with the following exceptions:

- Creating disksets
- Unmirroring file systems you cannot unmount
- Removing a UFS log from a file system that you cannot unmount
- Growing unmounted UNIX file systems

Before using DiskSuite Tool, you must first load the DiskSuite and DiskSuite Tool packages (`SUNWmd` and `SUNWmdg`) on a system running Solaris 2.4 and create the initial metadvice state database replica.

When you start DiskSuite Tool, you see a view of the system's actual DiskSuite configuration. You can use DiskSuite Tool to create a new configuration or manipulate an existing configuration.

DiskSuite and DiskSuite Tool Functionality

DiskSuite offers better performance, greater capacity, easier administration, and improved availability of disk storage on SPARC and x86 systems.

With DiskSuite, data availability and reliability are improved with up to three-way mirroring. You can mirror root (`/`), `/usr`, `swap`, or any file system. You can also mirror existing file systems and automatically replace failed components within a mirror using hot spare facilities. The mirrors feature includes optimized resyncing, which reduces the time required for synchronization.

The UNIX file system (UFS) logging facility provides faster local directory operations, speeds up reboots, and decreases synchronous disk writes by recording file system updates in a log before they are applied to the UNIX file system.

In addition, DiskSuite's disk striping can increase performance. Striping spreads data requests over multiple components. Concatenation and striping increase capacity by grouping several components into a single larger logical device.

The information about the configuration and state of all metadevices is preserved in a replicated state database. These databases are essential for the operation of DiskSuite.

Administration is simplified by the hot spare facility and the dynamic growth of metadevices and file systems.

Solstice DiskSuite 4.0 also features the creation of RAID (level 5) configurations and administration of metadevices within disksets. RAID provides recovery from disk failure in a more cost effective way than disk mirroring. A diskset is a grouping of two hosts and disk drives in which all the drives are accessible by both hosts.

Getting Help

If you have problems installing or using DiskSuite Tool, call the distributor from which you purchased the software and provide the following information:

- Your name and electronic mail address (if available)
- Your company name, address, and phone number
- The model and serial number of your system
- The release number of the operating system (for example, Solaris 2.4)
- Any additional information that will help diagnose the problem

Supported Software and Hardware

DiskSuite Tool may only be used on systems that are running Solaris 2.4 or a later version of the Solaris operating system. The system must have DiskSuite 4.0 or a later version installed. The Motif™ runtime package (SUNWmfrun) must be installed.

DiskSuite 4.0 runs on all SPARC and x86 systems that are running Solaris 2.4 or a later Solaris 2.x release.

The peripherals supported by DiskSuite and DiskSuite Tool include:

- IPI disk drives
- SCSI disk drives
- IDE disk drives (x86 only)

Note – The root file system cannot be mirrored on an IDE drive.

Disk Naming

DiskSuite Tool uses the normal disk naming convention for all drives. This naming is, *cn**tn**dn**sn*, where *cn* is the controller number, *tn* is the target, *dn* is the disk, and *sn* is the slice.

Note – However, be aware that on x86 systems, IDE drives do not have a target number. All the examples and instructions in this manual are from a SPARC system and show the target number.

Definition of Terms

The following are general definitions for DiskSuite and DiskSuite Tool:

Attach submirror - to add a submirror to an existing mirror. DiskSuite automatically resyncs the submirror with other submirrors.

Attach logging device - to add a logging device to an existing trans device.

Button - a graphical component on a window frame or in a dialog box that works by pressing it using the mouse.

Cascading menu - a submenu that provides selections that extend the parent selection on a pull-down or pop-up menu.

Component - the physical slice (partition) that is part of a metadvice.

Concatenated Stripe - a metadvice that is made up of both concatenated and striped components.

Concatenation - a metadvice created by sequentially mapping blocks on physical slices (partitions) to a logical device. Two or more physical components can be concatenated. The slices are accessed sequentially rather than interlaced (as with stripes).

Detach logging device - to remove a logging device from a trans device.

Detach submirror - to remove a submirror from a mirror.

Dialog box - a secondary window that conveys to or requests information from the user. Dialog boxes block the application until they are dismissed.

Diskset - an association of two hosts and group of disk drives in which all the drives are accessible by each host.

Drag and drop - using the mouse to select and move objects in a window or between windows. To drag and drop, press the middle mouse button, move the object to the desired location, and release the button.

Highlight - a visual cue signaling the current selection or current location of a selected object.

Hot spare - a component set up to automatically take over for a failed component of a mirrored or RAID metadvice.

Hot spare pool - a group of spare components that automatically replace failed components.

Icon - a small graphical image used to represent a window. Windows can be turned into icons or minimized to save room or clean up the workspace.

Interlace - the number of blocks on a component of a striped or RAID metadvice that can be accessed simultaneously with the same number of blocks from other components. The interlace value tells DiskSuite how much data to place on a component of a striped or RAID metadvice before moving on to the next component.

Logging - recording UNIX file system (UFS) updates in a log (the logging device) before the updates are applied to the UNIX file system (the master device).

Logging device - the component that contains the log for a trans device.

Master device - the component that contains the file system for a trans device.

Menu bar - a rectangular area at the top of the window, containing the titles of the pull-down menus.

Metadvice - a group of components accessed as a single logical device by concatenating, striping, mirroring, setting up RAID devices or logging the physical devices. Metadevices are sometimes called pseudo devices in UNIX domains.

Metadvice state database - information kept in nonvolatile storage (on disk) for preserving the state and configuration of metadevices.

Metadriver - a pseudo device driver that maps metadvice operations to operations on the metadvice components.

Mirror - a special type of metadvice made up of one or more other metadvice called submirrors. *See also* submirrors.

Mirroring - replicating all writes to a single logical device (the mirror) to multiple devices (the submirrors), while distributing read operations. This provides redundancy of data in the event of a failure.

Optimized resync - an update of only the submirror regions that are not in sync when the system reboots. The metadisk driver knows which regions of the submirrors are not identical (in sync) on all submirrors after a crash.

Partial resync - resyncing only a replaced component of a submirror or RAID device, rather than the entire submirror or RAID device.

Pop-up menu - a menu that provides no visual cue to its presence, but pops up when users perform a particular action. Pop-up menus are associated with a particular area of the workspace, such as the client area of an application.

Pull-down menu - a menu that is pulled down from a client application's menu bar.

RAID - an acronym for Redundant Arrays of Inexpensive Disks.

Replica - a copy of the state database. Keeping copies of the state database protects against the loss of state and configuration information. This information is critical to all metadvice operations.

Resync region - a division of a mirror that enables tracking changes by regions of the mirror, rather than over the entire mirror.

Resyncing - copying data from one submirror or RAID device to another after system crashes, submirror failures, or after a submirror is added. Resyncing ensures the data on all submirrors and on all RAID devices is identical.

Selection button - the mouse button used to make a selection.

State database - a dedicated portion of a disk reserved exclusively for the metadisk driver.

Stripe - similar to concatenation, except the addressing of the component blocks is interlaced on the slices (partitions), rather than sequentially. Striping is used to gain performance. By striping data across disks, multiple controllers can access data simultaneously.

Submirror - a metadvice that is part of a mirror. *See also* mirror.

Trans device - a special type of metadvice used for UFS logging. A trans device is composed of one or more other metadvice or components: a master device and optionally a logging device.

UFS - an acronym for the UNIX file system.

UFS logging - recording UNIX file system (UFS) updates to a log (the logging device) before the updates are applied to the UFS (the master device).

Installation and Setup



This chapter covers the basic Solstice DiskSuite Tool installation. Use the following table to locate specific information in this chapter.

<i>Local Installation</i>	<i>page 10</i>
<i>Remote Installation</i>	<i>page 11</i>
<i>Adding Packages</i>	<i>page 12</i>
<i>Viewing Installed Packages</i>	<i>page 21</i>
<i>Removing Packages</i>	<i>page 22</i>
<i>Path Requirements</i>	<i>page 23</i>
<i>Viewing AnswerBook Online Documentation</i>	<i>page 24</i>
<i>What to Do Next</i>	<i>page 24</i>

Note – You must have a CD-ROM drive installed on a machine somewhere on the network. Before creating any metadevices, the DiskSuite 4.0 software must be already installed and you must have created an initial metadevice state database and at least two replicas of it. If you have not created the initial state database replicas, either follow the procedure in Chapter 2 of the *Solstice DiskSuite 4.0 Administration Guide* or the instructions in Chapter 10, “State Database Replicas,” of this manual after installing DiskSuite and DiskSuite Tool.

For a full description of the Solaris 2.4 system software installation, refer to either *SPARC: Installing Solaris Software* or *x86: Installing Solaris Software*.

The basic steps to install software from CD-ROM include:

- Making sure you have enough space for the packages (DiskSuite Tool requires 10 Mbytes)
- Adding the software packages
- Setting the system path variables

Preparing for Installation

Instructions for installing DiskSuite Tool using a local or remote CD-ROM drive are given in the following subsections.

Local Installation

If you are installing DiskSuite Tool on a system with a CD-ROM attached, use the following instructions.

To prepare for a local installation:

- 1. Insert the DiskSuite Tool CD into the CD-ROM drive.**
- 2. Change directories to /cdrom/cdrom0 as follows:**

```
local% cd /cdrom/cdrom0
```

You are now ready to install the software on your local machine. Skip to the section “Adding Packages” on page 12 and follow the instructions provided.

Remote Installation

To prepare for a remote installation:

1. **On the remote machine, insert the DiskSuite Tool CD into the CD-ROM drive.**

2. **Put a line similar to the following into the `/etc/dfs/dfstab` file:**

```
share -F nfs -o ro -d "CD-ROM Directory" /cdrom/cdrom0
```

This line may be different, depending on how your system is networked.

3. **Export the `/cdrom/cdrom0` directory with the `shareall(1M)` command:**

```
remote# shareall
```

4. **On the local machine, log in as root and create the directory `/cdrom/cdrom0` (if it doesn't already exist):**

```
local% su
Password: root-password
local# mkdir -p /cdrom/cdrom0
```

You may choose another directory besides `/cdrom/cdrom0`.

5. **Mount the CD-ROM as follows:**

```
local# mount remote_machinename:/cdrom/cdrom0 /cdrom/cdrom0
```

You're now ready to install the software onto your local machine. Go to the section "Adding Packages" on page 12 and follow the instructions provided.

Adding Packages

1. Become root (if you haven't already).

```
local% su  
Password: root-password
```

2. Change to the directory on which the CD-ROM is mounted. (The section, "Mounting the CD-ROM," called this directory /cdrom/cdrom0. You may have chosen a different name.)

```
local# cd /cdrom/cdrom0
```

3. Run pkgadd to install packages.

```
local# pkgadd -d .
```

Note – If the `pkgadd` command is not in your current path, you must specify the full path to the command (`/usr/sbin/pkgadd`).

4. Choose the package you want to install.

pkgadd displays the available packages and prompts you to enter the number associated with a package. Select 1 to install Solstice DiskSuite only, 2 to install the Solstice DiskSuite Tool only (as illustrated in the example below), or 'all' to install both packages. The program loops until you press **q** to quit.

```
# pkgadd -d .
The following packages are available:
 1  SUNWmd      Solstice DiskSuite
      (all) 4.0,REV=1.0
 2  SUNWmdg     Solstice DiskSuite Tool
      (all) 4.0,REV=1.0

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: 2

Processing package instance <SUNWmdg> from
</cdrom/cdrom0/product>

Solstice DiskSuite Tool
(all) 4.0,REV=1.0
      Copyright 1994 Sun Microsystems, Inc. All Rights Reserved.
      Printed in the United States of America.
      2550 Garcia Avenue, Mountain View, California, 94043-1100 U.S.A.

... (miscellaneous copyright information)...

      (continued on following page)
```

```
Using </> as the package base directory.
## Processing package information.
## Processing system information.
    8 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing Solstice DiskSuite Tool as <SUNWmdg>

## Executing preinstall script.
## Installing part 1 of 1.
/usr/opt/SUNWmd/lib/X11/app-defaults/Metatool
/usr/opt/SUNWmd/lib/X11/bitmaps/browser_icon.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/browser_icon_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/concat_16.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/concat_16_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/concat_32.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/concat_32_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/concat_drag.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/concat_drag_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/controller_12.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/controller_16.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/controller_32.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/diskview_icon.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/diskview_icon_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/drop_none.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/drop_none_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/drop_not_ok.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/drop_not_ok_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/drop_ok.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/drop_ok_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/drop_site_selection.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/editor_icon.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/editor_icon_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/hsp_16.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/hsp_16_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/hsp_32.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/hsp_32_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/hsp_drag.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/hsp_drag_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/log_icon.xpm
```

(continued on following page)


```
/usr/opt/SUNWmd/lib/X11/bitmaps/log_icon_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/metadb_16.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/metadb_16_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/metadb_32.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/metadb_32_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/metadb_drag.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/metadb_drag_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/mirror_16.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/mirror_16_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/mirror_32.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/mirror_32_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/mirror_drag.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/mirror_drag_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/multi_drag.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/multi_drag_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/raid_16.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/raid_16_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/raid_32.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/raid_32_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/raid_drag.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/raid_drag_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/slice_16.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/slice_16_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/slice_drag.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/slice_drag_mask.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/target_12.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/target_16.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/target_32.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/trans_16.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/trans_16_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/trans_32.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/trans_32_insens.xpm
/usr/opt/SUNWmd/lib/X11/bitmaps/trans_drag.xbm
/usr/opt/SUNWmd/lib/X11/bitmaps/trans_drag_mask.xbm
/usr/opt/SUNWmd/lib/X11/uid/Metatool/Editor.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/EditorTmpl.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/concat.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/hsp.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/logs.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/md.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/mirror.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/physicalview.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/popups.uid
```

(continued on following page)

```

/usr/opt/SUNWmd/lib/X11/uid/Metatool/raid.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/slice.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/slicer.uid
/usr/opt/SUNWmd/lib/X11/uid/Metatool/stripe.uid
/usr/opt/SUNWmd/locale/C/help/metatool/howto/Howto
/usr/opt/SUNWmd/locale/C/help/metatool/howto/cat.create.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/catinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/catstripe.del.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/cflog.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/cflog.show.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/clnup.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/clpse.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/col.assign.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/col.map.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/commit.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/confirm.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/delete.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/diskinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/diskview.displaying.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/diskview.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/duplicate.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/eval.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/exit.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/expand.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/fail.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/find.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/help.help.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.about.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.addspares.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.associating.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.browser.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.browsing.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.create.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.deletespares.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.enable.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.getinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.replacing.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.setfilters.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hsp.statuscheck.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/hspinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/info.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/interlace.h.hlp

```

(continued on following page)

```
/usr/opt/SUNWmd/locale/C/help/metatool/howto/load.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/log.to.file.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/maps.show.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mapselect.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/md.browser.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/md.filters.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mddbinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mded.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mdinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mds.browsing.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mirror.create.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mirror.existing.fsys.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mirror.onlinebackup.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mirror.root.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mirror.statuscheck.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mirror.swap.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mirror usr.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mirrorinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/mousing.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/pan.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/panics.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/phy.map.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/problist.display.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/problist.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/putaway.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/raid.create.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/raidinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/save.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/sdb.create.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/sdb.rem.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/sdb.status.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/selectall.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/slicebrowser.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/slicefilters.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/sliceinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/slices.add.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/slices.browsing.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/slices.enable.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/slices.getinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/slices.replace.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/slices.showinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/spares.adding.h.hlp
```

(continued on following page)

```

/usr/opt/SUNWmd/locale/C/help/metatool/howto/spares.deleting.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/spares.enabling.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/spares.replacing.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/spares.status.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/stripeinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/stripes.add.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/stripes.create.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/stripes.info.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/stripes.rem.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/submirr.attach.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/submirr.detach.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/submirr.enable.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/submirr.offline.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/submirr.online.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/submirr.rep.comp.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/submirr.setoptions.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/trans.create.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/trans.rem.logging.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/trans.sharelog.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/trans.status.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/trans.umtfilesystems.logging.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/transinfo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/ufslog.setup.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/undo.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/howto/unmirror.filesys.h.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/Reference
/usr/opt/SUNWmd/locale/C/help/metatool/reference/catinfo.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/cflog.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/confirm.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/dev.suppl.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/dialogs.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/diskinfo.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/diskview.about.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/diskview.areas.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/diskview.menus.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/find.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/fs.log.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/gloss.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/growfs.cautions.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/growfs.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/guide.avail.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/guide.capacity.r.hlp

```

(continued on following page)

```
/usr/opt/SUNWmd/locale/C/help/metatool/reference/guide.labels.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/guide.perform.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/help.help.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/hspbrowser.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/hspfilters.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/hspinfo.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/load.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/log.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mapselect.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mdbbrowser.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mddbinfo.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mded.areas.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mded.buttons.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mded.men.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mdfilters.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mdname.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mirror.examples.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mirror.naming.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/mirrorinfo.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/pan.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/phy.to.log.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/problist.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/raidinfo.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/save.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/security.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/slicebrowser.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/slicefilters.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/sliceinfo.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/spares.cautions.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/spares.naming.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/spares.pools.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/spares.states.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/stripeinfo.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/stripes.cautions.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/sysfiles.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/targets.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/trans.naming.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/transinfo.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/reference/ufs.space.reqmts.r.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/Topics
/usr/opt/SUNWmd/locale/C/help/metatool/topics/cat.stripe.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/cflog.t.hlp
```

(continued on following page)

```

/usr/opt/SUNWmd/locale/C/help/metatool/topics/colors.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/commit.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/dialogs.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/diskinfo.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/diskview.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/find.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/gethelp.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/gloss.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/growfs.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/help.help.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/intro.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/key.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/load.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/log.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/mdt.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/mded.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/mds.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/mirror.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/object.create.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/phy.map.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/problist.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/putaway.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/raid.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/save.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/sdb.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/slicebrowser.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/slicefilters.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/slices.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/spares.about.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/stripes.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/templates.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/trans.t.hlp
/usr/opt/SUNWmd/locale/C/help/metatool/topics/warning.t.hlp
/usr/opt/SUNWmd/man/man1m/metatool.1m
[ verifying class <none> ]
/usr/opt/SUNWmd/sbin/metatool
[ verifying class <sparc> ]

```

Installation of <SUNWmdg> was successful.

(continued on following page)

```
The following packages are available:
 1  SUNWmd      Solstice DiskSuite
                   (all) 4.0,REV=1.0
 2  SUNWmdg     Solstice DiskSuite Tool
                   (all) 4.0,REV=1.0

Select package(s) you wish to process (or 'all' to process
all packages). (default: all) [?,??,q]: q
```

5. Respond with a **y** to any prompts about changing modes on directories.
pkgadd installs the DiskSuite Tool files in the /usr/opt/SUNWmd directory.
pkgadd does not overwrite any software included with the standard Solaris 2.4 release.

Viewing Installed Packages

You can confirm that the DiskSuite Tool software and AnswerBook have been installed by using the `pkginfo` command:

```
demo$ pkginfo
SPARCompile SUNWC++      SPARCompilers 2.0  C++ 3.0
system      SUNWaccr     System Accounting, (Root)
system      SUNWaccu     System Accounting, (Usr)
SPARCompile SUNWacomp    SPARCompilers 2.0  Sun C 2.0
system      SUNWarc      Archive Libraries
system      SUNWast      Automated Security Enhancement Tools
system      SUNWaudio    Audio applications
system      SUNWbcp      Binary Compatibility
system      SUNWbnur     Networking UUCP Utilities, (Root)
system      SUNWbnuu     Networking UUCP Utilities, (Usr)
system      SUNWmdg      Solstice DiskSuite Tool
system      SUNWmd       Solstice DiskSuite
...
```

The `-l` option gives detailed information about packages:

```
demo$ pkginfo -l SUNWmdg
PKGINST:  SUNWmdg
NAME:      Solstice DiskSuite Tool
CATEGORY:  system
ARCH:      all
VERSION:   4.0,REV=1.0
BASEDIR:   /
VENDOR:    SunSoft, a Sun Microsystems, Inc. Business
DESC:      SunSoft's Solstice DiskSuite Tool
PSTAMP:    machineABC950120190546
INSTDATE:  Jan 23 1995 04:10
VSTOCK:    258-4905-10
HOTLINE:   Please contact your local service provider
STATUS:    completely installed
FILES:     295 installed pathnames
           6 shared pathnames
           19 directories
           1 executables
           5402 blocks used (approx)
```

Removing Packages

If you decide to remove the DiskSuite Tool package, remove the installed files using the `pkgrm(1M)` command. For example (as superuser), type:

```
# pkgrm SUNWmdg
```

Note – Do not remove DiskSuite Tool files using the `rm(1)` command. Using `pkgrm` is the only valid way to remove these files.

Path Requirements

After installing the software, you must set the environment variables `PATH` and `MANPATH`.

The general requirements are as follows:

- Add `/usr/opt/SUNWmd/sbin` either before or after `PATH`.

If you do not set this path, you may get other versions of the software.

- Add `/usr/opt/SUNWmd/man` either before or after your `MANPATH`.

If you do not set this path, you could get man pages for the wrong release, or no man pages at all.

You can set these paths by using an editor to change your `$HOME/.profile` or `~/.cshrc` file, as follows.

If you installed DiskSuite Tool into the default directory, then:

- If you're using the Bourne shell, your `$HOME/.profile` file should have lines like this:

```
PATH=/usr/opt/SUNWmd/sbin:$PATH
MANPATH=/usr/opt/SUNWmd/man:$MANPATH
export PATH MANPATH
```

- If you're using the C shell, the `~/.cshrc` file should have lines that look something like this:

```
set path = (/usr/opt/SUNWmd/sbin $path)
setenv MANPATH /usr/opt/SUNWmd/man:$MANPATH
```

Viewing AnswerBook Online Documentation

To view the AnswerBook® online documentation:

1. **Type** `/installpath/SUNWabmdg/answerbook` **and press Return.**
installpath is the path displayed at the end of installation. In the installation example shown previously, the path on a local system is:

```
$ /opt/SUNWabmdg/answerbook
```

2. **A title screen is displayed. Follow the online instructions to use the AnswerBook online documentation.**

If you are viewing the AnswerBook product on a network, you need to specify the entire path to the system on which the AnswerBook product is installed. For example:

```
$ /net/server/opt/SUNWabmdg/answerbook
```

What to Do Next

Now that the DiskSuite Tool software is set up on your system, you can begin to use the functionality provided. Read Chapter 3, “Overview of DiskSuite Tool,” to familiarize yourself with the product.

Overview of DiskSuite Tool



This chapter provides a high-level overview of the DiskSuite Tool features and explains the parts of the graphical interface. Use the following table to locate specific information.

<i>DiskSuite Description</i>	<i>page 25</i>
<i>DiskSuite Tool's User Interface</i>	<i>page 29</i>
<i>Starting DiskSuite Tool</i>	<i>page 30</i>
<i>Metadevice Editor Window</i>	<i>page 32</i>
<i>Disk View Window</i>	<i>page 38</i>
<i>Information Windows</i>	<i>page 42</i>
<i>Browsers</i>	<i>page 46</i>
<i>Dialog Boxes</i>	<i>page 52</i>
<i>Configuration Log Window</i>	<i>page 53</i>
<i>Problem List Window</i>	<i>page 54</i>
<i>Accessing and Using Help</i>	<i>page 55</i>
<i>DiskSuite Recovery and Guidelines</i>	<i>page 56</i>

DiskSuite Description

This subsection gives a brief overview of the DiskSuite functionality that is accessed using DiskSuite Tool. You should review the *Solstice DiskSuite 4.0 Administration Guide* for a complete description of the underlying functionality.

DiskSuite provides a metadisk driver which is implemented as a set of loadable, pseudo device drivers. The metadisk driver uses other physical device drivers to pass I/O requests to and from the underlying devices. The metadisk driver resides between the file system interface and the device driver interface and interprets information from both above and below.

A brief overview of the primary elements of the metadisk driver is given in the following subsections. The elements include:

- Metadevices
- Concatenation and striping
- Mirroring (mirrors and submirrors)
- UFS logging (trans)
- Hot spare pools
- Disksets
- RAID devices

Other parts of the DiskSuite software that are explained in the following subsections include:

- Metadevice state database replicas
- Expanding mounted file systems

Metadevices

Metadevices are the basic functional unit of the metadisk driver. After you create metadevices, you can use them like physical disk slices. These logical devices can be made up of one or more component slices. You can configure the component slices to use a single device, a concatenation of stripes, or stripe of devices.

Metadevices can provide increased capacity, higher availability, and better performance. To gain increased capacity, you create metadevices that are either concatenations, stripes, or RAID devices. Mirroring, UFS logging, and RAID devices provide higher availability. Disk striping can help performance.

Metadevices are transparent to application software and to component and controller hardware.

Metadevices can be configured from IPI and SCSI devices on all SPARC systems and on SCSI and IDE devices on all x86 systems.

Concatenations and Stripes

Each metadvice is either a concatenation or a stripe of component slices. Concatenations and stripes work much the same way the `cat(1)` program is used to concatenate two or more files together to create one larger file. When slices are concatenated, the addressing of the component blocks is done on the components sequentially. The file system can use the entire concatenation. Striping is similar to concatenation, except the addressing of the metadvice blocks is interlaced on the components, rather than addressed sequentially.

Use the information in Chapter 4 to create concatenations and stripes.

Mirroring

DiskSuite supports mirroring to as many as three separate metadvicees. This enables the system to tolerate single-component failures with two-way mirroring and double failures with three-way mirroring. Mirroring can also be used for online backups of file systems.

To set up mirroring, you create a mirror. A mirror is a special type of metadvice made up of one or more other metadvicees. Each metadvice within a mirror is called a submirror.

Complete information about mirroring is provided in Chapter 5.

UFS Logging

The UFS logging facility included with DiskSuite provides faster local directory operations, speeds up reboots, and decreases synchronous disk writes by safely recording file system updates in a log before they are applied to the UFS file system.

UFS is the standard Solaris file system. UFS file systems are created when Solaris is installed or by users with the `newfs(1M)` command.

Use the information in Chapter 6 to create UFS logs.

Hot Spares

DiskSuite's hot spare facility automatically replaces failed submirror or RAID components, provided that a suitable spare component (one as large as the component that is being replaced) is available and reserved. Hot spares are temporary fixes, used until failed components are either repaired or replaced. Hot spares automatically re-establish the original redundancy following a failure. This reduces the time data is vulnerable to a subsequent failure.

Chapter 7 provides information on how to create, use, and manipulate hot spares using DiskSuite Tool.

Disksets

DiskSuite's diskset feature lets you set up one or two host machines and a group of disk drives in which all of the hosts in the set are connected to all the drives in the set.

Chapter 8 contains information about administering metadevices in disksets.

RAID Devices

DiskSuite supports redundant arrays of inexpensive disks (RAID) metadvice configurations. These are level 5 RAID devices. When implemented as metadevices, RAID is composed of three or more physical slices; each slice is referred to as a column. The slice of a RAID metadvice can be increased by concatenating additional slices to the metadvice.

RAID metadevices are covered in Chapter 9.

Metadvice State Database Replicas

State database replicas provide the nonvolatile storage necessary to keep track of configuration and status information for all metadevices, mirrors, trans devices, and hot spares. The replicas also keep track of error conditions that have occurred.

Chapter 10 provides detailed information about state database replicas.

Expanding Mounted File Systems

You can expand all mounted UNIX file systems except root (/) by using DiskSuite Tool. The expansion can be performed without bringing down the system or performing a backup.

Mounted file systems can be expanded up to the new size of the metadvice on which the file system resides.

Note – Once you have expanded a file system, you cannot shrink it.

Expanding mounted file systems is covered in Chapter 11.

DiskSuite Tool's User Interface

The DiskSuite Tool user interface is based on Motif™. The window border, however, is OPEN LOOK®. Thus the mouse behavior inside the DiskSuite Tool windows is that of Motif. There are noticeable differences in the look and feel between Motif applications and OPEN LOOK. The following list details the differences.

- The mouse model is different.

Table 3-1 DiskSuite Tool Mouse Model

Button	Function
Left	Enables you to select objects with a single click. By holding down the Control key and clicking the left button, you can select multiple objects. By holding down the Control key and clicking the left button, you can deselect objects that are selected. Additionally, you can drag objects by holding down the left button.
Middle	Drags selected objects. You can drop the object on an appropriate target. For example, concatenations can be mirrored using the drag and drop method. If a target is not appropriate, the international “no” sign displays while the cursor is over the target.
Right	Displays pull-down menus when the cursor is pointing at any title in the menu bar. Inside the canvas of the DiskSuite Tool windows or inside objects, the pop-up menu for a object is displayed.

- Pressing the Return key on DiskSuite Tool dialog boxes applies the default action, which may not be the desired action. The default action is in a highlighted button.
- Scrollbars look different under Motif than under OPEN LOOK. However, the left mouse button is still used to scroll up and down.
- Pushpins are not part of the Motif interface. Instead, the title bar contains the window menu on the left side rather than a pushpin. The choices on the menu include Close, Full Size, Move, Resize, Properties, Back, Refresh, Stick, and Quit.
- Small graphic symbols are used as cursors and in Dialog Boxes. An important cursor symbol is a small international “no” sign (a circle with a diagonal line), which indicates an action is not allowed. Some Dialog Boxes appear with question marks and others with warnings.
- The online help utility can be accessed using the Help menu on the menu bar of any DiskSuite Tool window or dialog box.

Starting DiskSuite Tool

DiskSuite Tool (`metatool(1M)`) offers a graphical user interface to the functionality provided by the DiskSuite 4.0 software package. DiskSuite Tool can be used to create and administer mirrors, concatenations, stripes, metadvice state database replicas, UFS logs, hot spares, and RAID devices.

All metadevices are represented in a graphical interface based on the Motif window manager, which is displayed in OPEN LOOK windows on Solaris 2.4.

The advantages of DiskSuite Tool over the previous command-line interface are the graphical representation of all metadevices and the convenient drag-and-drop interface.

Note – Before you can commit metadevices created using DiskSuite Tool, a minimum of three metadvice state database replicas must exist on the system. Either use the instructions in the *Solstice DiskSuite 4.0 Administration Guide* or in Chapter 10, “State Database Replicas,” in this manual to create the metadvice state database replicas.

You must be root to use the DiskSuite Tool. To start the DiskSuite Tool, enter:

```
# metatool
Initializing metatool... Done.
Discovering drives and slices... Done.
Discovering database replicas... Done.
Discovering hot spare pools... Done.
Discovering concat/stripes...Done.
Discovering RAID devices... Done.
Discovering mirrors... Done.
Discovering trans devices... Done.
Updating mount and swap information... Done.
```

The messages shown above are displayed each time you invoke `metatool`. This feedback tells you the product is working properly. If you invoke the command with the `-s` option, additional messages about the metaset will be displayed.

An overview of the windows, browsers, dialog boxes, and menus is given in the following subsections.

Metadevice Editor Window

When invoked, `metatool` displays the Metadevice Editor window, as shown in Figure 3-1. The Metadevice Editor window is where most manipulation and creation of metadevices takes place. From this window, you will access the other pieces of DiskSuite Tool.

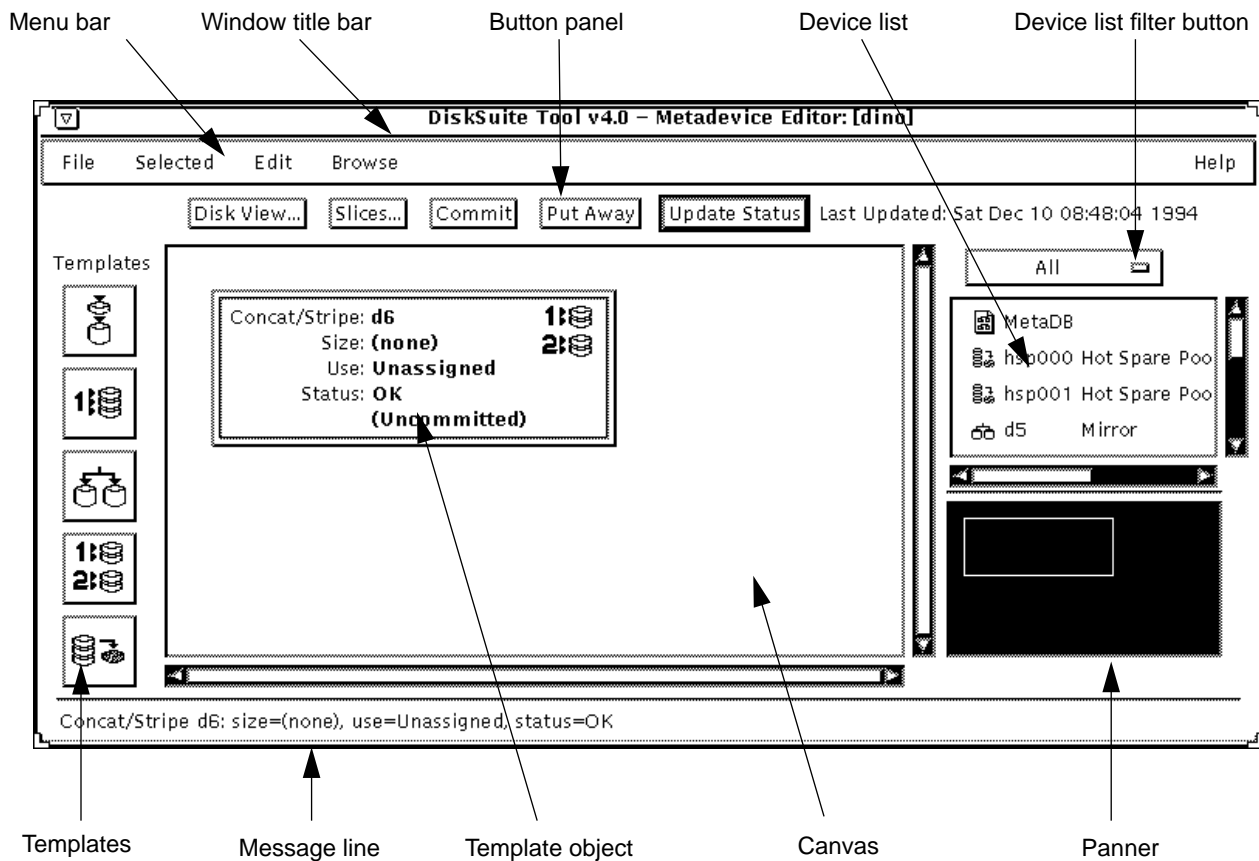


Figure 3-1 DiskSuite Tool Metadevice Editor Window

An explanation of the parts of the Metadevice Editor window follows:

- **Menu bar** – The menu bar offers pull-down menus titled File, Selected, Edit, Browse and Help. Point to the title and press the left button to see the choices available. Some choices may be grayed out at certain times, if the functionality provided is not appropriate. A listing of the menu choices and functions are given in Table 3-2.

Table 3-2 Metadevice Editor Window Menu Choices

Menu	Choices	Functions
File	Save to File Restore from File Exit	Save uncommitted configurations to a file Display configurations from a file Exit DiskSuite Tool
Selected	Info Expand Collapse Evaluate Put Away Commit	Displays appropriate information window. Displays entire selected device. Collapses the selected device view completely. Displays errors and warning in the Configuration Log and presents report. Returns selected (committed) metadevice to device list. Writes changes made to selected metadevices to all state database replicas and may update the <code>/etc/vfstab</code> file. (Note: The above choices apply to any selected object on the canvas. Objects are selected when you click on them.)
Edit	Undo Last Undo All Duplicate Delete Create Select All Clean Up Canvas	Will undo most previous actions. Will undo any actions performed since the last time you clicked on Commit. Makes copies of the selected objects. Removes the selected objects. Opens templates for Trans, RAID, Mirror, Concat/Stripe, and Spare Pools. Selects all the objects on the canvas. Rearranges the objects on the canvas.
Browse	Disk View Slices Metadevices Hot Spare Pools Find Configuration Log Problem List	See page 38 See page 46 See page 46 See page 46 See page 49 See page 53 See page 54 (Note: See the specified page for illustrations and details about the menu items.)
Help	On Help On Window	Displays help on the DiskSuite Tool Help system. Displays help for the Metadevice Editor Window.

- **Window title bar** – The title of the DiskSuite Tool window is displayed in the title bar. For example, “DiskSuite Tool v4.0 - Metadevice Editor [dino]” is displayed in Figure 3-1. The title of the window is followed by the name of the system where `metatool` is running. It is important to remember to check the host name before configuring metadevices.
- **Button panel** – The selections along the button panel provide a quick way to view the Disk View window, Slice Browser window, to Commit changes, to Put Away metadevices, and to Update Status. Click on a button to invoke an action. To select an object on the canvas to Commit or Put Away, first point to the object and select it by clicking the left mouse button, then point to the appropriate button and click the left button.

The Commit and Put Away functions work on all the objects that are selected. There are two ways to select multiple objects on the Metadevice Editor canvas, as follows:

- Draw a rubber band box around the objects. This is done by pointing to a corner of the canvas and pressing the left button. Then drag the cursor to encompass the objects.
- Point to the top of the first object and click the left button. Press and hold down the Control key and point to the top of other objects and click the left button.

Changes made to disks are only recognized when you click on the Update Status button.

The button panel selections are shown in Figure 3-2.

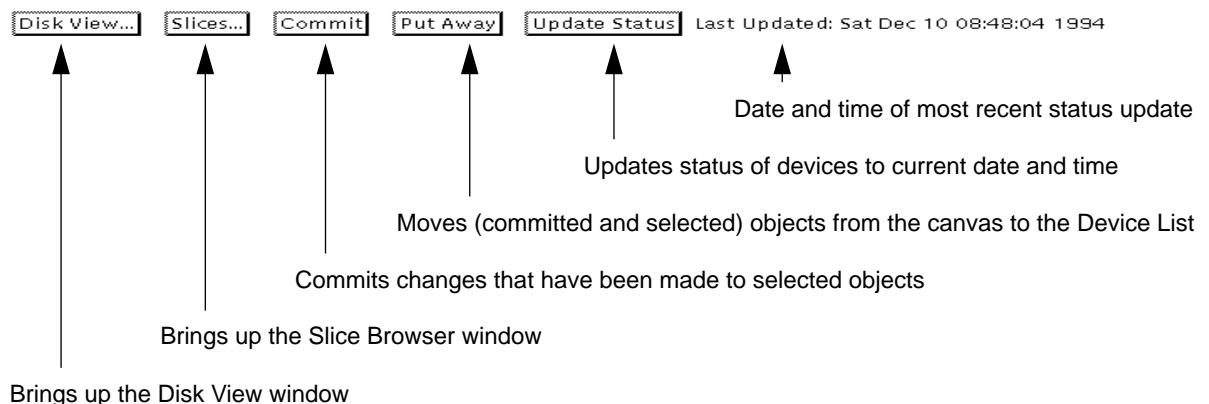


Figure 3-2 Button Panel Selections

- **Templates** – This region contains graphical representations that are the source for new objects. These objects can be used as templates when creating trans devices (UFS logging), RAID devices, mirrors, concatenations (stripes), or hot spare pools. To move the objects to the canvas, you either click once on a template or drag it to the canvas. The template objects are shown in Figure 3-3.

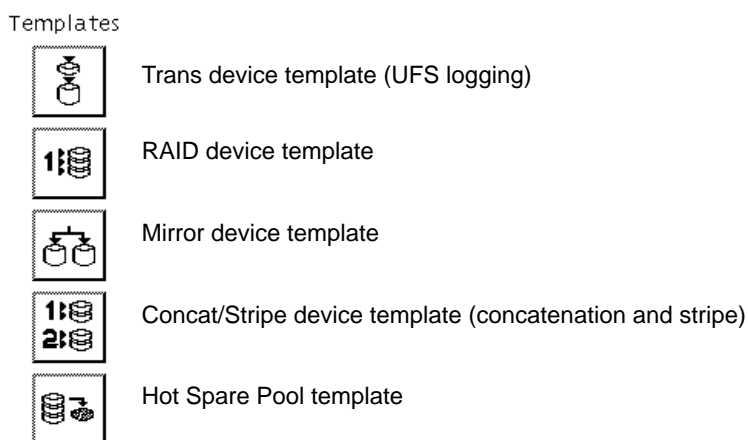


Figure 3-3 Template Objects

- **Message line** – When the cursor is over an area of the Metadevice Editor window, a message about that area is displayed in the message line. For example, when the cursor is pointed at the RAID template, the following message appears: “Click or drag to create a new RAID device.” A complete list of messages is given in Appendix B, “DiskSuite Tool Messages.”
- **Template object** – Clicking on one of the templates opens a template object. Slices from the Slice Browser window and metadevices from the device list can be dropped into the template objects. For example, Concat/Stripe objects can be dropped into a Mirror template to create mirrors.
- **Canvas** – This region is where graphical objects are displayed and manipulated. The canvas has horizontal and vertical scrollbars for scrolling the visible portion of the canvas. The panner (in the lower right corner of the metatool window) also can be used to scroll the visible portion of the canvas. You can drag objects from the Disk View window, the device list, or

from the Templates to the canvas. Click on the objects on the canvas to select them. When an object is selected, the border becomes dark or is highlighted in orange. You can drag the object to a new location on the canvas.

- **Device list** – The names of the mirrors, concatenations, RAIDs, trans devices, concatenations, stripes, hot spare pools, and the metadvice state database are shown in the device list. To move these objects to the canvas for manipulation, click on the item and drag it to the canvas or double-click on the object. Horizontal and vertical scrollbars allow scrolling the visible portion of the list.

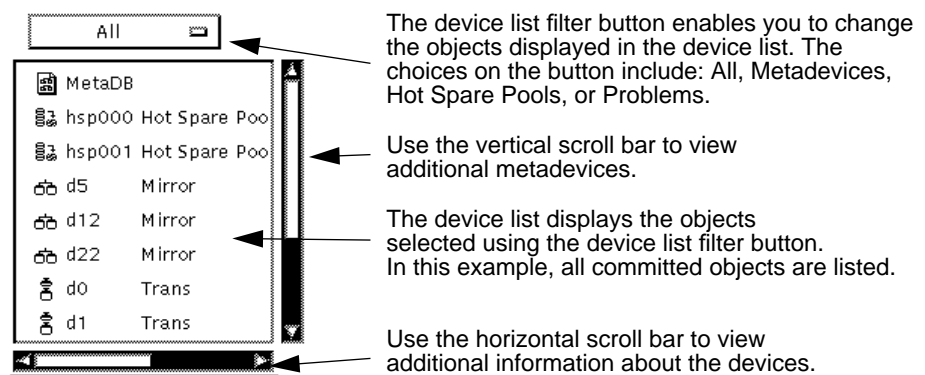


Figure 3-4 Device List

On color monitors, the objects in the device list are displayed in different colors when a problem has occurred. The following colors are used:

- Critical - Red
- Urgent - Orange
- Attention - Yellow

There is a discussion about each of the types of errors (critical, urgent, and attention) at the end of each chapter that describes a metadvice. For example, refer to “Checking the Status of Concatenations and Stripes” on page 91 for a listing of the errors for concatenations and stripes.

Gray scale monitors operate the same as color monitors, however the colors are displayed in gray scales. On monochrome monitors, the status rectangle is black. You must scroll the device list horizontally to view the status associated with the devices.

- **Device list filter button** – This button offers a filter that changes the information displayed in the device list. The choices on the filter include: All, Metadevices, Hot Spare Pools, and Problems.
- **Panner** – The panner is a miniature view of the canvas and shows small representations of the objects currently displayed on the canvas. The black rectangle in the panner represents the portion of the canvas currently displayed on the screen. To change the view, point inside the white region and click the left button. Another method of changing the view is to point to the black rectangle, press and hold down the middle button, and drag the view area to a new location.

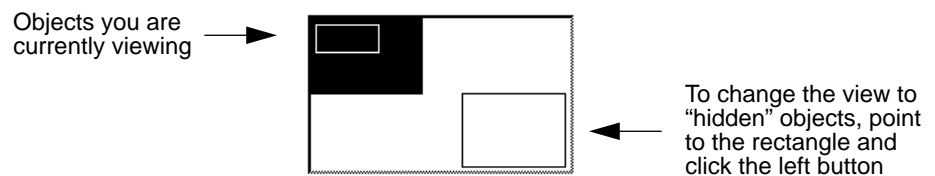


Figure 3-5 Panner Region

Disk View Window

The Disk View window shows how metadevices (logical devices) correspond to physical devices. It also serves as a drag-and-drop source for slices and a drag-and-drop target for metadevices. The Disk View window is shown in Figure 3-6.

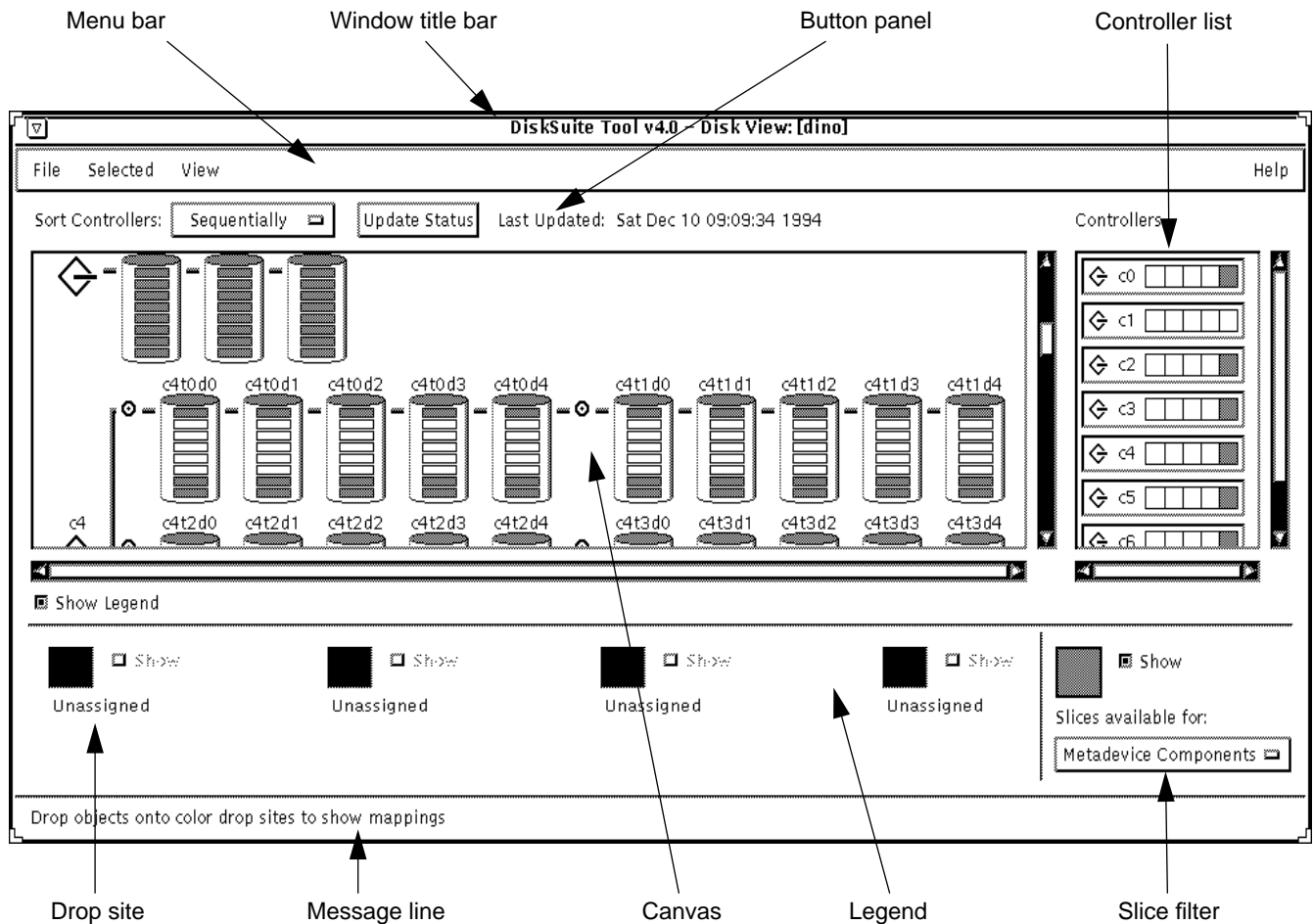


Figure 3-6 Disk View Window

An explanation of the parts of the Disk View window follows:

- **Menu bar** – The menu bar offers pull-down menus titled File, Selected, View, and Help. Point to the menu option and press the left button to see the choices available. Some choices may be grayed out at certain times.

Table 3-3 Disk View Window Menu Choices

Titles	Choices	Functions
File	Close	Closes the Disk View window.
Selected	Info Associations Expand Collapse	Displays information window. Displays a logical devices list. Enlarges view (or size) of the selected device. Restores view (size) of the selected device. (Note: The above choices apply to any selected devices on the canvas. Devices are selected when you click on them.)
View	Device Mappings Device Status Expand All Collapse All View All Controllers Put Away All Controllers Drop Site Colors Replica Slice	Toggles Disk View to display logical-to-physical mappings. Toggles Disk View to display status of the slices instead of mappings. Expands all object views on the canvas. Collapses all object views on the canvas. Displays all controllers on the canvas. Removes all controllers from the canvas. Enables selection of colors for slices. Displays replica slice.
Help	On Help On Window	Displays help on the DiskSuite Tool Help system. Displays help for the Disk View Window.

- **Window title bar** – The title is followed by the name of the system where DiskSuite Tool is running. Be sure to check the host name before configuring devices.
- **Button panel** – There are two buttons in the button panel region, Sort Controllers and Update Status. The Sort Controllers button specifies the controller sort order on the canvas. The values for the pulldown menu are Sequentially, Drop Site 1, Drop Site 2, Drop Site 3, Drop Site 4, Available Slices, Critical, Urgent, and Attention. Sequentially is the default. The Update Status button updates the status on all parts of the DiskSuite Tool.

- **Controller list** – All controllers that are part of the current context are shown in the controller list. The graphical representations of the controllers in the list act as toggle switches. You open or close a controller by pointing to the object in the list and clicking the left button.
- **Drop site** – Any metadvice (or group of metadvice) can be dropped on the color drop sites. Dropping the object on a specific color causes that color to be used for the mapping. The drop site shows the color assignments for logical-to-physical mappings.

The colors used for each of the four drop sites in the Disk View window may be changed while DiskSuite Tool is running. Any changes made to the selection of these colors will not be maintained when you exit DiskSuite Tool. Eight colors are available for drop sites on color systems: blue, green, magenta, cyan, purple, medium sea green, firebrick, and tan.

To change the colors used for the four drop sites, choose the option “Drop Site Colors” from the View pull-down menu from the Disk View menu bar. This option is a cascading menu, and has four additional cascading menus, one for each drop site. Each of the menus for each drop site contains a menu of eight colors. Each available color is in the form of a toggle button. The currently selected color is indicated by the selected toggle. Release the cursor over the desired color. All references to this color throughout the Disk View window will be updated to use the selected color.

The drop site is illustrated in Figure 3-7.

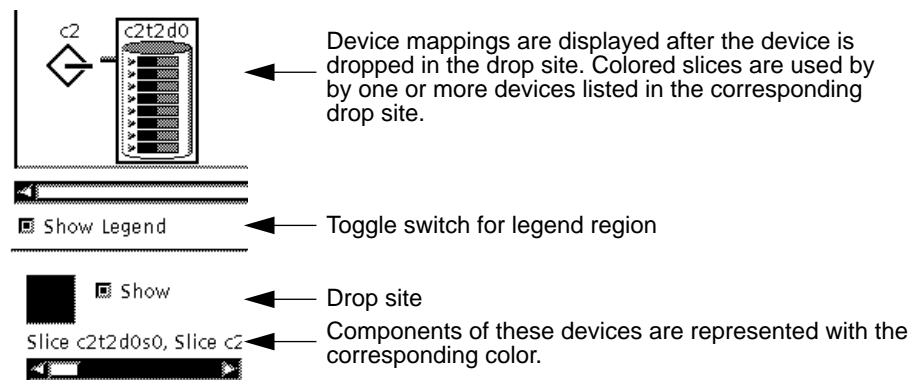


Figure 3-7 Drop Site Objects

- **Canvas** – The physical devices and mappings are displayed on the canvas. The canvas has horizontal and vertical scrollbars for scrolling the visible portion of the canvas. To select a disk on the Disk View canvas, click on the top of the disk. To select a slice, click inside the slice rectangle. After the object is selected you can drag the object to a template on the Metadevice Editor canvas and add or replace slices in that template.

The canvas is also a destination for drag and drop. When devices are dropped on the canvas they take on the next available color. If all dropsites are in use, a window is displayed that enables you to select a dropsite. Also, if any object is selected on the editor canvas and the Disk View window is invoked, the objects will automatically take on the color of the next available dropsite.

The graphical representations of objects on the Disk View canvas are shown in Figure 3-8.

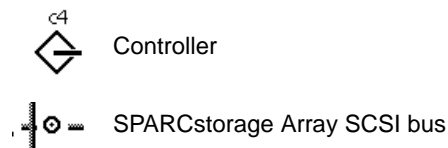


Figure 3-8 Disk View Objects

- **Legend** – The legend region of the Disk View window can be turned on and off using the Show Legend toggle button. On color systems, the legend contains four color drop sites that provide color cues for mappings. Each color can be hidden or exposed using the Show toggle button to the right of each color box. On monochrome systems, only one drop site is available, which is black.



Figure 3-9 Legend Region

- **Slice filter** – A fifth component of the legend region is the “Slices available for” option menu. The menu enables you to filter the slices to view those available to be Metadevice Components, Hot Spares, Replicas, Trans Logs, or Anything. The default value is Metadevice Components.
- **Message line** – When the cursor is over an area of the Disk View window, a message about that area is displayed in the message line. For example, when the cursor is pointed at the canvas, the following message appears: “Drop objects onto color drop sites to show mappings.” A complete list of messages is given in Appendix B, “DiskSuite Tool Messages.”

Information Windows

Several information windows are associated with DiskSuite Tool. These information windows include:

- Disk Information Windows (see page 43)
- Slice Information Windows (see page 44)
- Concat Information Windows (see page 87)
- Stripe Information Windows (see page 89)
- Mirror Information Windows (see page 137)
- Trans Information Windows (see page 163)
- Hot Spare Information Windows (see page 188)
- RAID Information Windows (see page 214)
- Metadevice State Database Information Windows (see page 231)

The Disk Information and Slice Information windows are discussed in this overview. Overviews of the other information windows are presented in the chapters where the information is discussed.

Disk Information Window

By pointing to a disk on the Disk View canvas and pressing the right button, a menu enables you to bring up an information window. This window, the read-only Disk Information window, provides information about a disk and its slices.

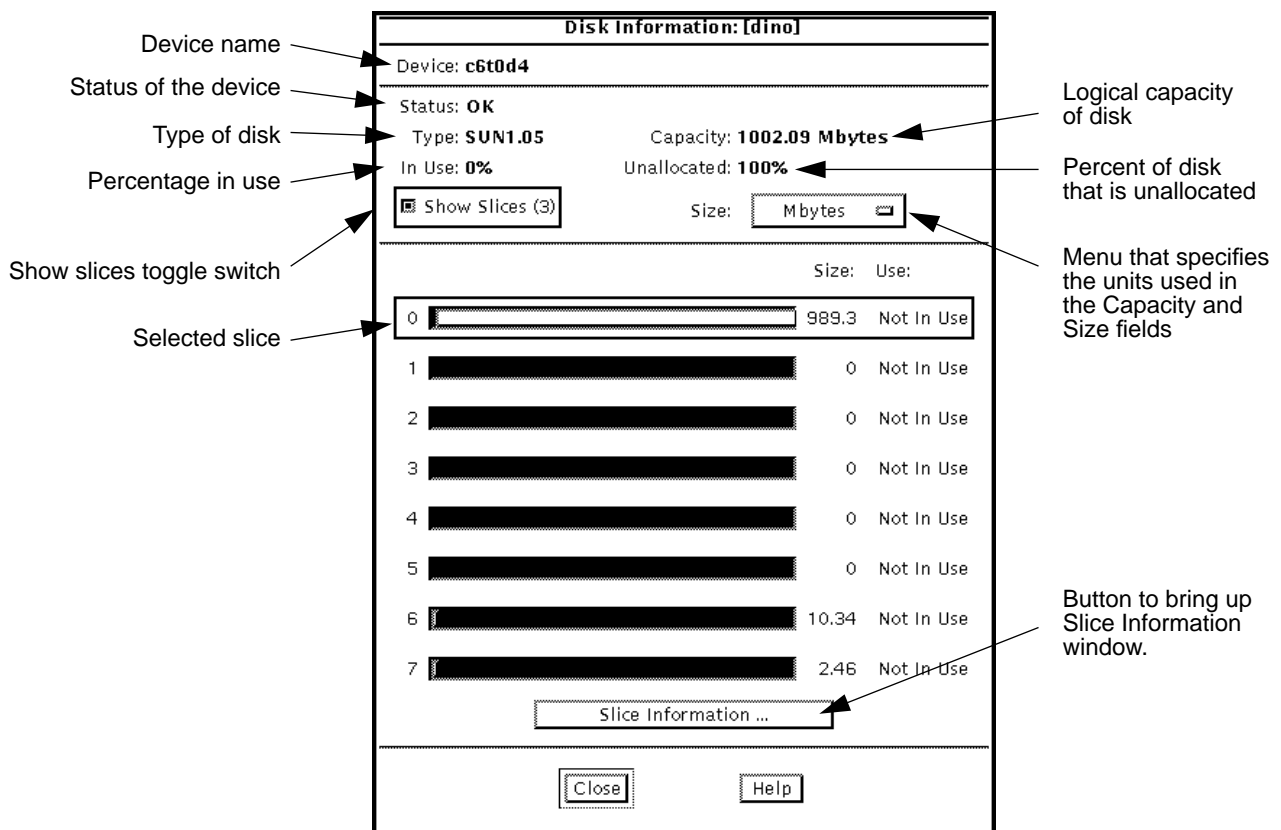


Figure 3-10 Disk Information Window

Table 3-4 lists the functionality provided by the Disk Information screen.

Table 3-4 Disk Information Screen Functionality

Field	Functions
Device name	The physical device name, for example, <code>c6t0d4</code> .
Status	The status is reported as OK, or Failed.
Type	The type of the disk as listed in the <code>/etc/format.dat</code> file, for example SUN0535, SUN1.05, or DEFAULT.
In Use	The percentage of the disk that is currently in use as a metadvice, metadvice state database replica, or a mounted file system.
Capacity	The logical capacity of the disk.
Unallocated	The percentage of the disk available for use.
Size	The area where you can change the size units represented in the capacity field and the information under the Size column in the Slice region. Choices include: Gbytes, Mbytes, Kbytes, Sectors, and Cylinders. The default is Mbytes.
Show Slices	A radio button that expands and collapses the slice view. The number of non-zero slices on the disk is shown in parentheses on the button.
Slice Info	A button that brings up the Slice Information window for each selected slice. Point to the slice area and click the left button to select a slice. To select multiple slices, either press and hold down the Control key while pointing to the slices and clicking the left button or hold down the left button and drag the cursor over slices.

Slice Information Window

The slice information window displays information about a specific slice. There are three ways to display this window:

- Select a slice on the Disk Information window by pointing to it and pressing the left button. Then point to the Slice Information button and click the left button.
- Point to a slice of a disk displayed on the Disk View window's canvas. Press and hold down the right button to bring up the menu for the slice and select the Info option.

- Point to a slice inside any metadvice displayed on the Metadvice Editor's canvas. Press and hold down the right button to bring up the menu for the slice and select the Info option.

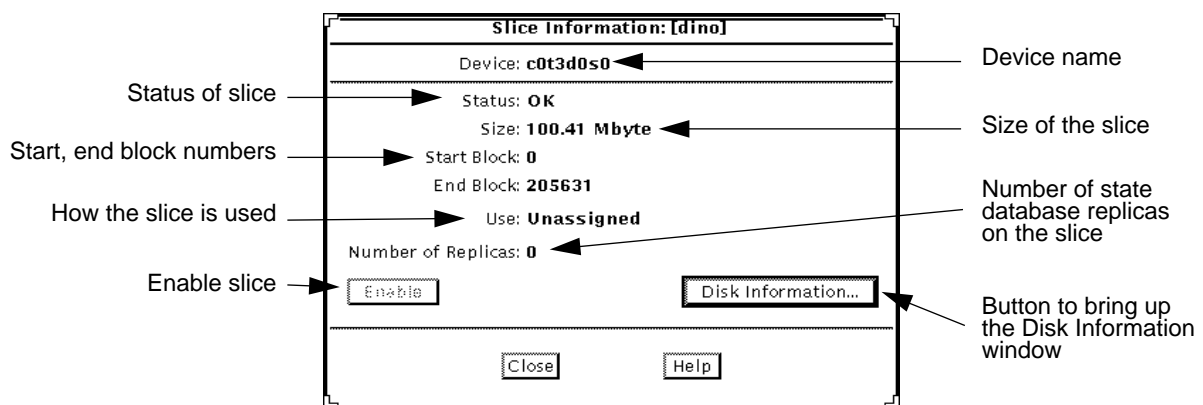


Figure 3-11 Slice Information Window

Table 3-5 gives an explanation of the Slice Information screen.

Table 3-5 Slice Information Screen Functionality

Field	Functions
Device name	The physical device name, for example, c0t3d0s0.
Status	The status is reported as OK, Resyncing, Enabled, Critical, Spared, Urgent, or Attention.
Size	The size of the slice.
Start block	The block on which the slice begins. If the slice has a label, there will be additional information about the label in this field.
End block	The block on which the slice ends.
Use	The current use of the slice, for example, file system or swap. If the use is hot spare, a Show HSPs button is available on the right side of the Use field. This button opens a dialog that shows a list of Hot Spare Pools with which the slice is associated.
Enable	This button enables the slice. The button is available only if the data on the slice is replicated in a mirror or RAID. Also, the slice could be a hot spare that is currently out of service.
Disk Information	Opens the Disk Information window.

Browsers

Four browsers can be accessed from the Browse menu on the Metadevice Editor window. These include:

- Slice Browser
- Metadevice Browser
- Hot Spare Pool Browser
- Find Browser

The first three browsers provide the same functionality: enabling you to view all information about the slices, metadevices, and hot spare pools and drag these objects to the Metadevice Editor's canvas for manipulation. The find browser is used to locate an object in the metadevice editor's window, or locate the device associated with a specified mount point.

The first three browsers have many similar characteristics. The only noticeable differences are found in some of the information displayed and in the Set Filter windows.

The Slice Browser window is shown in Figure 3-12.

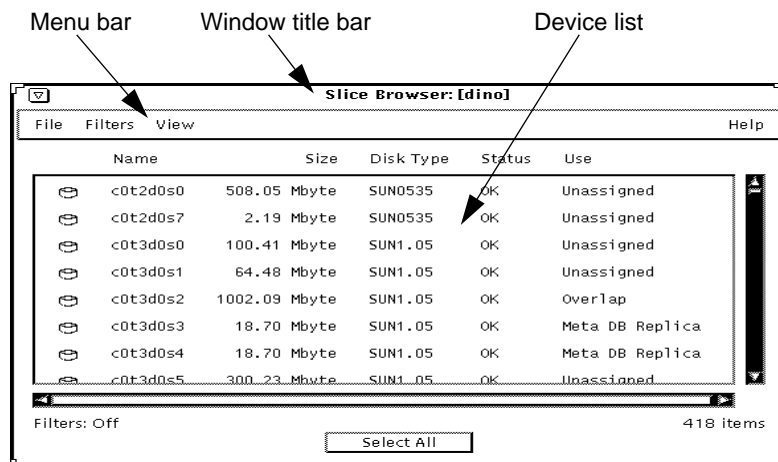


Figure 3-12 Slice Browser Window

The Slice, Metadevice, and Hot Spare Pool browsers all have the same window title bar and choices on the menu bar. The File menu enables you to exit the browser. The Filters menu enables you to set the filters and turn them on and off. The View menu enables you to change the order in which information is displayed in the device list. However, there are some subtle differences on the dialog boxes used to set the filters.

The device list varies in the following ways:

- **Slice Browser Device List** – To view additional information about the slices listed here, point to a slice and double-click the left mouse button. The Slice Information window displays information about the slice and provides access to the Disk Information window. The Slice Browser device list contains the information shown in Table 3-6.

Table 3-6 Slice Browser Device List Information

Field	Function
Name	The physical device name, for example, c0t3d0s0.
Size	The size of the device.
Disk Type	The type of the disk as listed in the <code>/etc/format.dat</code> file, for example, SUN0535, SUN1.05, or DEFAULT
Status	Reported as OK, Resyncing, Enabled, Critical, Spared, Urgent, or Attention.
Use field	Contains one of the following values: Unassigned, Trans Log, Trans Master, MetaDB Replica, Component, File System currently mounted on slice, Overlap, or Hot Spare.

- **Metadevice Browser Device List** – To view additional information about the metadevices listed, point to a metadevice and double-click the left mouse button. An information window that contains information about the

type of metadvice is displayed (and the object is opened on the Metadvice Editor's canvas). The Metadvice Browser device list contains the information shown in Table 3-7.

Table 3-7 Metadvice Browser Device List Information

Field	Function
Name	Presents the metadvice is represented as <i>dn</i> , where the default value for <i>n</i> is a number in the range 0 to 127.
Size	The size of the metadvice.
Type	The type is reported as either Trans, Concat/Stripe, Mirror, or RAID.
Status	The status is reported as OK or Failed.
Use	The use is either Unassigned, Submirror of <i>dn</i> , name of a file system, Master of <i>dn</i> or Trans Log.

- **Hot Spare Pool Device List** – To view additional information about the hot spare pools listed, point to a hot spare pool and double-click the left mouse button. The Hot Spare Information window is displayed, showing a list of the metadvice that have an association with the hot spare pool. It also shows information about the disks in the pool. The Hot Spare Pool device list contains the information shown in Table 3-8.

Table 3-8 Hot Spare Pool Device List Information

Field	Function
Name	The name of the hot spare pool is displayed as <i>hspnnn</i> , where <i>nnn</i> is a number in the range 000 to 999.
Min Size	The size of the smallest slice in the hot spare pool.
Max Size	The size of the largest slices in the hot spare pool.
Spares	The number of hot spares in the pool.
Spares in Use	The number of hot spares currently in use.
Status	The status is OK, Empty (there are no slices in the hot spare pool), Broken (if all slices in the hot spare pool are broken), or Attention (if one or more slices are in use).

- **Find Browser** – To locate an object inside the Metadvice Editor window, select the Find choice and enter the name or mount point in the appropriate text fields of the Find window (see Figure 3-13). If the object is anywhere on

the canvas, it is placed in the upper left corner. The object will become the current selection (any previously selected objects will be deselected.) If the object is in the Device List, it is opened and placed in the upper left corner of the canvas. The text fields are not case sensitive. Wildcard character support includes both the asterisk (*) and question mark (?). The asterisk matches zero or more characters and the question mark matches one character.

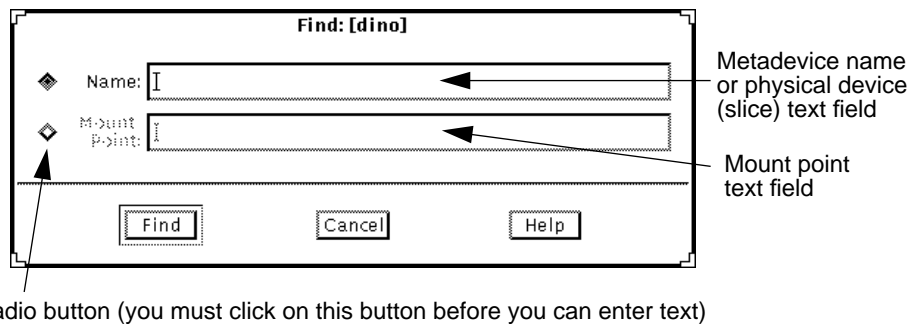


Figure 3-13 Find Window

Setting Browser Filters

The first three browsers previously described have configurable Filter windows that are available using the Set Filters choice in the Filter item on the menu bar. The filters are used to change the way information is displayed in the device list. The Slice Filters window is shown in Figure 3-14.

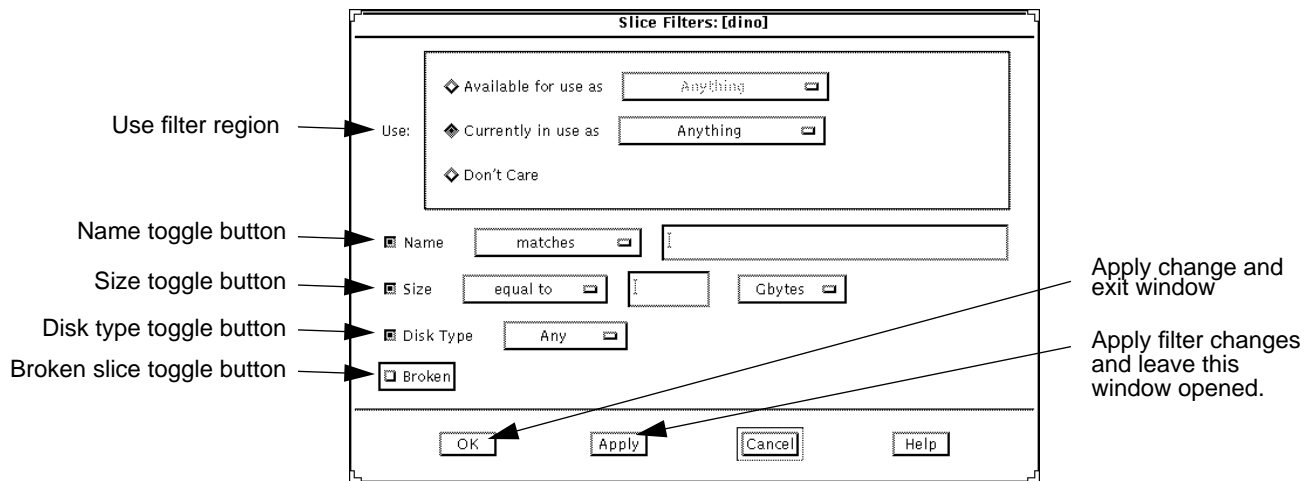


Figure 3-14 Slice Filter Window

An explanation of the Slice Filter window items is given in Table 3-9.

Table 3-9 Slice Filter Window Items

Window Region	Function
Use	Three toggle buttons in this region enable you to filter the display to only show the following: Available for use as – The menu selections here include: Anything, Hot Spare, Replica, Metadevice Component or Trans Log. Anything is the default. Currently in use as – The menu selections here include: Anything, File System, Swap, Replica, Metadevice, Hot Spare, Component, or Trans Log. Anything is the default. Don't care – The filter is ignored.
Name toggle button	Turning on the name toggle button enables you to specify a device name. The two choices on the menu enable you to search for names that match or do not match. Wildcard character support includes both the asterisk (*) and question mark (?), which match any number of characters and any single character respectively. Matches is the default.
Size toggle button	Turning on the size toggle button enables you to specify a size for the filter. The menu button selections include: from (a field is added for specifying a “to” range), greater than, less than, equal to, and not equal to. The default is equal to. A size menu button enables you to specify Gbytes, Mbytes, Kbytes, and Sectors.
Disk Type toggle button	Turning on the Disk Type toggle button enables you to select the types of disks you wish to have displayed in the browser. The menu always enables you to select Any, but the other selections depend on the types of disks attached to your system.
Broken toggle button	Searches for only slices that have a “broken” status.

Accessing Objects in the Browsers

All objects in the device list of any of the three browsers can be moved to the Metadevice Editor's canvas for manipulation. To move the objects:

1. **Point to the appropriate slice, metadevice, or hot spare pool on the device list and click the left mouse button. The object is selected.**

2. **Point to the selected object. Press and hold down the middle mouse button.**
3. **Drag the object to the Metadevice Editor's canvas.**

Alternatively, pointing to the object and double-clicking causes the information window to pop up and the object to appear on the canvas. If the object is a component of another metadevice, the parent metadevice will be displayed.

Dialog Boxes

DiskSuite Tool displays feedback to users via four different types of dialog boxes at various times. Dialog boxes block user input to DiskSuite Tool until they have been dismissed.



Caution – Read and understand the dialog boxes before responding. You can inadvertently lose data.

An example of a warning dialog box is shown in Figure 3-15.

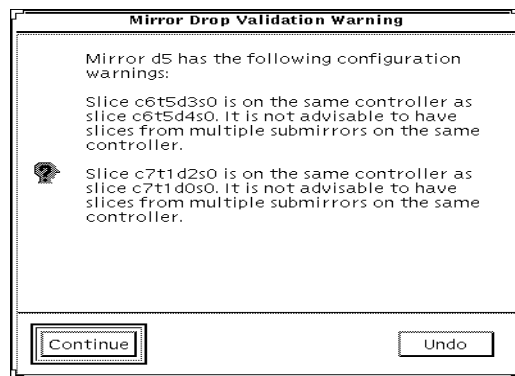


Figure 3-15 Example Dialog Box

The types of dialog boxes and the information they display are shown in Table 3-10.

Table 3-10 Dialog Boxes Information

Type	Information Presented
Error	When you attempt to perform an action that will result in an error, an error dialog box appears with a notification of the error.
Warning	When you attempt to perform an action that results in a warning, you are given the opportunity to cancel the action. Appendix B, “DiskSuite Tool Messages” offers a listing of the error messages and the corrective action.
Confirmation	These provide a way for you to confirm an action that has been selected. These will appear when an action you initiated cannot be undone. The message string in each dialog varies according to the operation.
Information	These provide a helpful message. These dialog boxes appear with a large “i” on the left side of the message. There is only one button on the bottom of this dialog: OK.

Configuration Log Window

The Configuration Log window provides a history of all top-level DiskSuite operations.

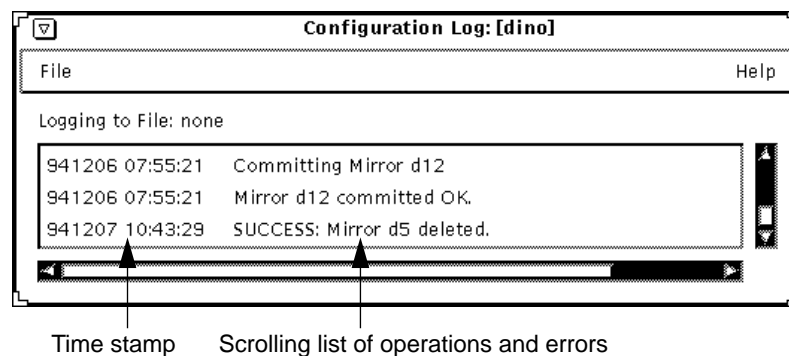


Figure 3-16 Configuration Log Window

Selections on the File menu enable you to clear the scrolling list, logging the messages to a user-designated file, and closing the window. Double-clicking on an entry in the list brings up the information dialog window for the device and opens the device on the Metadevice Editor's canvas.

Problem List Window

The Problem List window contains a scrolling list of the current metadevice problems. The list does not provide a history of the problems that have been encountered. The list is updated each time DiskSuite Tool learns of a change in status. Each item on the list is given a time stamp.

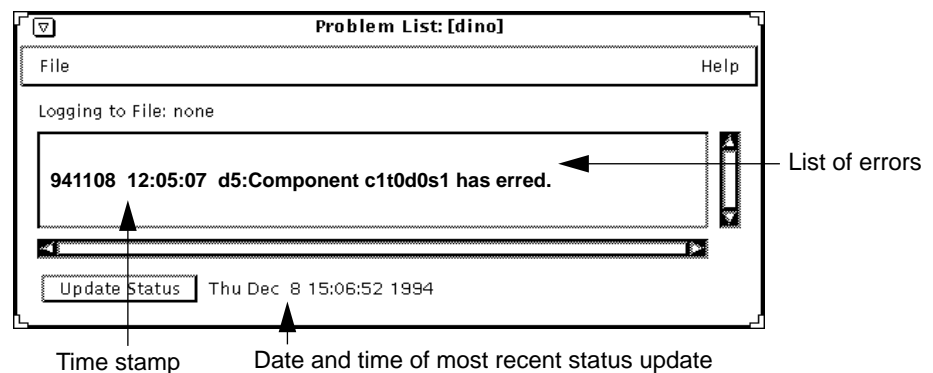


Figure 3-17 Problem List Window

Selections on the File menu enable you to log the messages to a user-designated file and close the window. The Update Status button updates the status of devices to the current date and time. The text field on the right side of the button displays the date and time of the most recent update.

Double-clicking on an entry in the list brings up the information window for the device and opens the device on the Metadevice Editor's canvas.

Accessing and Using Help

An important part of the DiskSuite Tool is the online help utility. The help program provides detailed information about the DiskSuite Tool and the functionality it provides. The DiskSuite Tool help utility is shown in Figure 3-18.

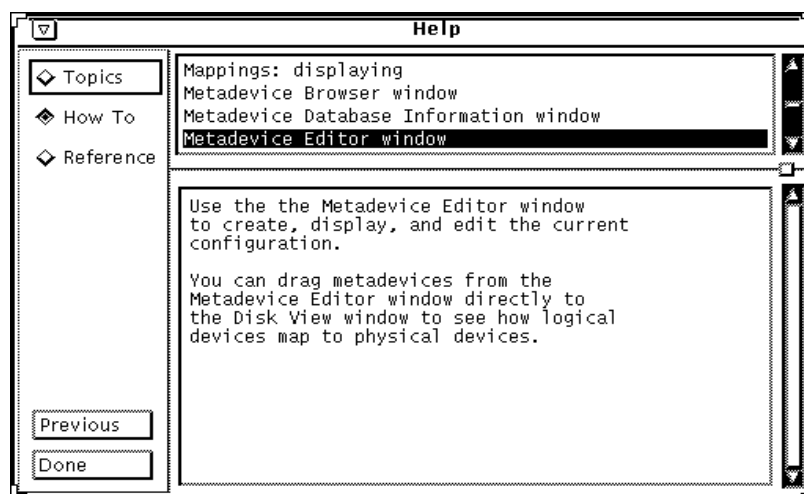


Figure 3-18 DiskSuite Tool Help Utility

To access the help utility from any of the DiskSuite Tool windows:

1. **Point to Help on the menu bar.**
2. **Press the left mouse button.**
3. **Select either “On Help” or “On Window” from the menu.**
4. **The Help window is displayed.**

To access the online help from a DiskSuite Tool dialog box:

1. **Click on the Help button.**
2. **The Help window is displayed.**

The Help titles displayed in the top window pane identify the list of subjects available for each level of help.

The Help text displayed in the bottom window pane describes information about using the current menu or command.

Use the scrollbars to the right of each pane to scroll through the help information displayed.

On the left side of the Help utility are buttons used to find information and navigate through the help system. The buttons are described in Table 3-11.

Table 3-11 DiskSuite Tool Help Buttons

Button	Click On This Button To...	Then...
Topics	Display a list of conceptual topics	Click on a title in the top window pane to view the accompanying help text.
How To	Display a list of step-by-step procedures	
Reference	Display a list of references	
Previous	Return to the last-accessed help topic	The help viewer returns to the previous screen.
Done	Exit the online help system	The help system is closed.

DiskSuite Recovery and Guidelines

Not all information about DiskSuite has been reproduced in this manual. You should use the *Solstice DiskSuite 4.0 Administration Guide* in conjunction with this manual.

Certain topics that are important to the operation of DiskSuite are included in the following subsections.

Recovery From Failed Boots

Because DiskSuite Tool supports mirroring and logging the root (/), swap, and /usr file systems, special problems can arise when you are booting your system, either if a hardware failure occurs or if you have not followed all instructions.

The primary problems that can happen include:

- Metadevice state databases that contain invalid (stale) data
- Improper /etc/vfstab entries

- Boot device failure

Appendix C of the *Solstice DiskSuite 4.0 Administration Guide* presents examples of such problems and provides possible solutions.

Configuration Guidelines

Consider the following factors building DiskSuite configurations using DiskSuite Tool:

- Performance
- Availability
- Capacity
- Security
- Compatibility
- Labeled slices

Chapter 12 of the *Solstice DiskSuite 4.0 Administration Guide* gives more information.

Concatenating and Striping



This chapter discusses how to use the Solstice DiskSuite Tool (`metatool(1M)`) to create metadevices that consist of either concatenated or striped disk slices. Refer to Chapter 3, “Overview of DiskSuite Tool,” for information on using DiskSuite Tool.

Use the following table to locate specific information in this chapter.

<i>Overview of Concatenating and Striping</i>	<i>page 59</i>
<i>Using Concatenations and Stripes</i>	<i>page 61</i>
<i>Hardware and Software Considerations</i>	<i>page 62</i>
<i>Defining Metadevice Configurations</i>	<i>page 63</i>
<i>Replacing Failed Components</i>	<i>page 80</i>
<i>Clearing Concatenations and Stripes</i>	<i>page 84</i>
<i>Concat Information Window</i>	<i>page 87</i>
<i>Stripe Information Window</i>	<i>page 89</i>
<i>Checking the Status of Concatenations and Stripes</i>	<i>page 91</i>

Overview of Concatenating and Striping

Each metadevice is either a concatenation or a stripe of component slices. Concatenations and stripes work much the way the `cat(1)` program is used to concatenate two or more files together to create one larger file.

When slices are concatenated, the component blocks are addressed sequentially. The file system can access the entire concatenation.

Note – You can use a concatenated or striped metadvice for any file system with the exceptions of root (/), swap, /usr, /var, /opt, or any other file system accessed during a Solaris upgrade or installation.

Figure 4-1 illustrates the concatenation of three 327-Mbyte components. The logical block address ranges are listed below the drives. The physical block address range for each of the three drives would be 0 to 669695. When concatenated, the logical block address range is sequential from 0 to 2009087. In this illustration the disk labels are not part of the logical block addresses.

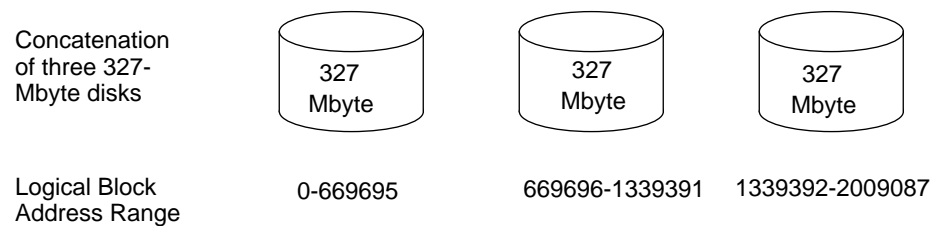


Figure 4-1 Concatenation of Three 327-Mbyte Drives

Striping is similar to concatenation, except the addressing of the metadvice blocks is interlaced on the components, rather than addressed sequentially. When stripes are defined, an interlace size is specified as a number, for example, 8, 16, 32. The numbers are specified as kilobytes, megabytes, or (512-byte) sectors. If the size is not specified, it defaults to 16 Kbytes. This value tells DiskSuite how much data is placed on a component before moving to the next component of the stripe.

Because data is spread across a stripe, you gain increased performance as reads and writes are spread across multiple disk arms. Also, concurrent I/O requests may use different disk arms. This may be true of concatenation as well.

In Figure 4-2, three 327-Mbyte components are used to illustrate a stripe of component slices. The sector size is 512 bytes. The interlace value is 8 Kbytes (or 16 512-byte sectors). The same logical address range as shown for the concatenation in Figure 4-1 (0 to 2009087) applies to a stripe of the same component configuration. However, in striping, the logical block addresses on each component are alternated according to the interlace size specified. In this illustration the disk labels are not part of the logical block addresses.

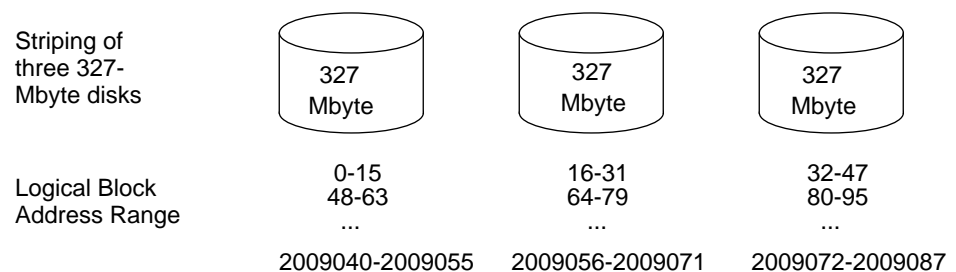


Figure 4-2 Striping of Three 327-Mbyte Drives

Figure 4-2 further illustrates how improved performance is gained through striping. For example, if a 1-Mbyte request were issued to this configuration, the data would be spread across the three components and the component arms on all components would be used to retrieve the data concurrently.

Using Concatenations and Stripes

Concatenating disk slices enables you to create a single metadvice with large capacity. This provides a solution for the space limitation a single slice poses. The logical disk block addresses are allocated sequentially to concatenated disk slices.

You can use a concatenated or striped metadvice for any file system with the exception of root (/), swap, /usr, /var, /opt, and any other file system accessed during a Solaris upgrade or installation.

Striping performs the same function as concatenation, enabling you to create a single metadvice with large capacity. But striping differs from concatenation because the logical disk block addresses are allocated in an interlaced fashion, which can improve performance.

The interlace value for striping is user defined. If the interlace size is not specified on the Stripe Information window, the size defaults to 16 Kbytes. A performance gain occurs when the I/O request is larger than the interlace size, because more disk arms are used to retrieve data or multiple requests spread over more disk arms.

DiskSuite Tool supports concatenations consisting of a single-component slice. This enables you to temporarily define a single-component configuration in anticipation of adding more components in the future.

Note – Another use of a single-component configuration is to enable mirroring of that component. Root (/), /usr, swap, and any other file system used when the system is booting must be in single-component configurations.

Concatenations and stripes are given the metadvice name `dn`, where *n* is a number in the range 0 to 127.

After the metadvice is configured, it can be used just as if it were a physical slice. As with physical slices, a file system can be created on the metadvice. Most UNIX disk utilities will work normally on metadevices, with the exception of `format(1M)`.

In addition, all usual file system operations can be performed on a metadvice. The following list offers examples of file system operations:

- Mount (`mount(1M)`) the metadvice on a directory
- Unmount (`umount(1M)`) a mounted metadvice
- Copy files to the mounted metadvice
- Read and write files from and to the mounted metadvice
- Run `ufsdump(1M)` and `ufsrestore(1M)` on the metadvice

A drawback of concatenations, stripes, and concatenated stripes is that the loss of a single physical slice can cause the loss of service to the entire metadvice. This can be solved through mirroring or RAID and the use of hot spares. These three topics are discussed in subsequent chapters.

Hardware and Software Considerations

Both hardware and software considerations affect concatenations and stripes.

The only software consideration involves determining the interlace size when you build a stripe.

The hardware considerations include component size, number of controllers, component geometry, and the I/O load on the bus.

Refer to Chapter 4 of the *Solstice DiskSuite 4.0 Administration Guide* for additional information on the hardware and software considerations.

Defining Metadevice Configurations

The following sections show how to define concatenations, stripes, and concatenated stripes using DiskSuite Tool.

Concatenated Metadevices

Concatenation can help solve the single slice capacity limitation. For instance, concatenations enable a single database to exist on a single metadevice spanning multiple physical slices.

Figure 4-3 shows two physical components and a metadevice. The physical components are `/dev/dsk/c6t5d4s0` and `/dev/dsk/c7t0d0s0` and the metadevice is `/dev/md/dsk/d6`. As illustrated in Figure 4-3, a metadevice is a “virtual disk” and can be used like a physical disk slice.

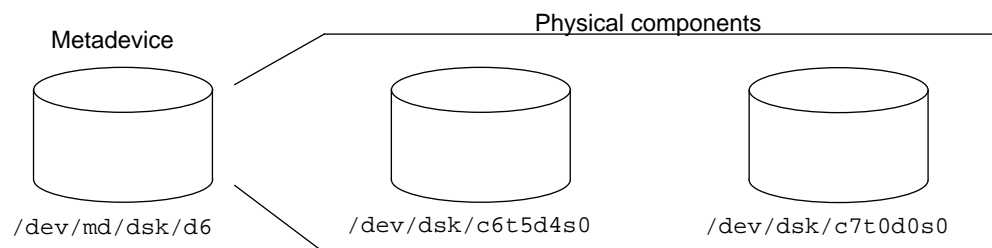


Figure 4-3 Concatenated Metadevice

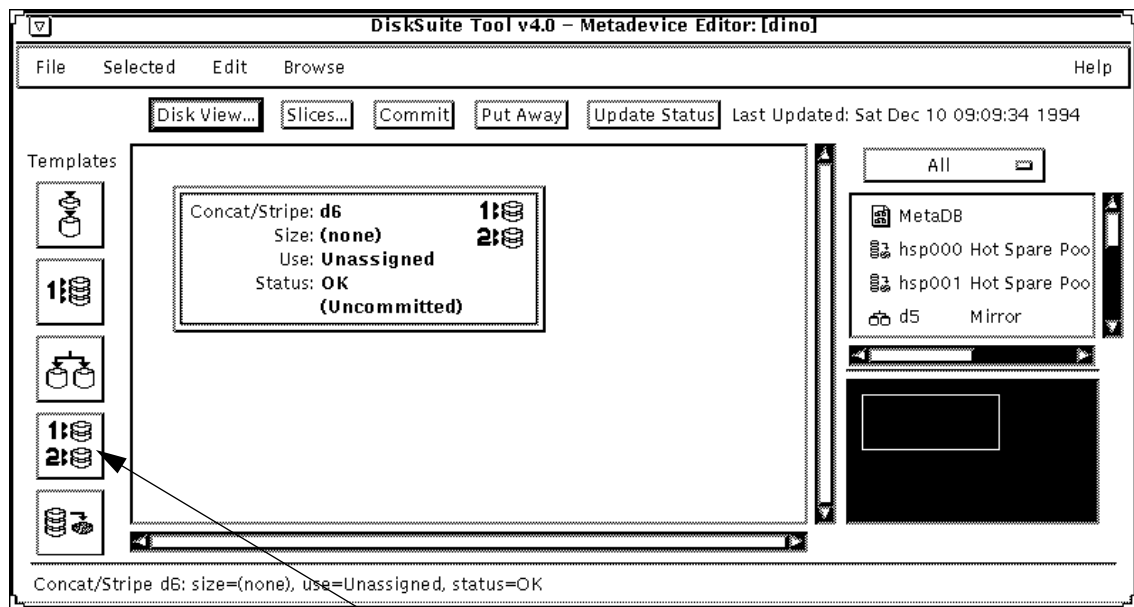
To set up a concatenation of the two physical components as shown in Figure 4-3, using DiskSuite Tool, perform the following steps:

1. Start DiskSuite Tool.

```
# metatool
```

2. Open a Concat/Stripe template on the Metadevice Editor canvas.

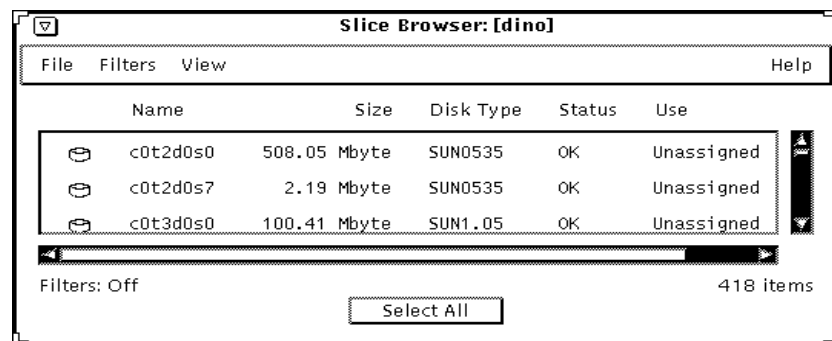
Click on the Concat/Stripe template. An unassigned and uncommitted Concat/Stripe template is opened on the Metadevice Editor canvas. When you point to the template, the message line shows the metadevice name, size, use, and status.



Click here to open the Concat/Stripe template

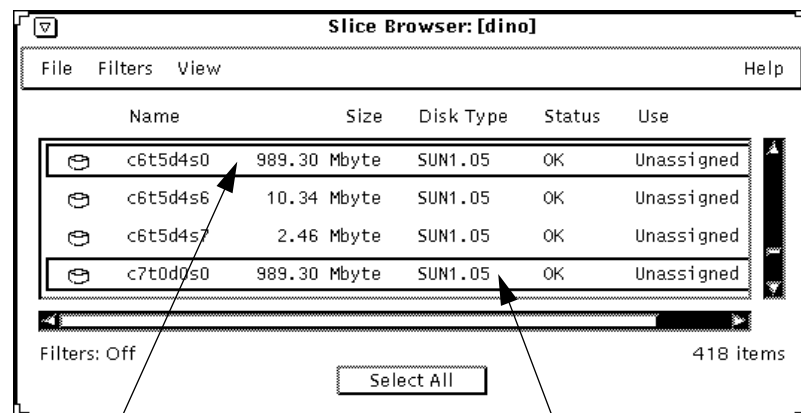
3. Open the Slice Browser.

Click on Slices on the button panel.



4. Select the slices that will be concatenated.

Scroll through the list of the devices on the Slice Browser and locate the two slices that will be concatenated. Click on the first slice that will be concatenated, then hold down the Control key and click on the second slice.



Click here to select the first slice.

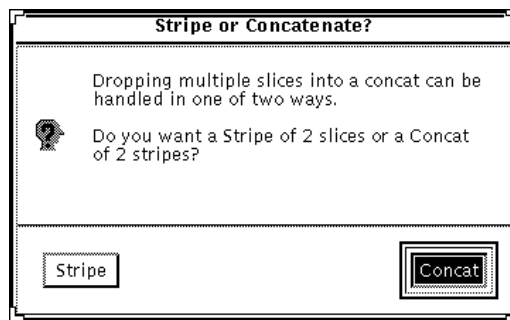
Hold down the Control key and click here to select the second slice.

5. Drag the two slices into the Concat/Stripe template.

Point to one of the two selected slices, press the middle button, and drag the slices to the Concat/Stripe d6 template.

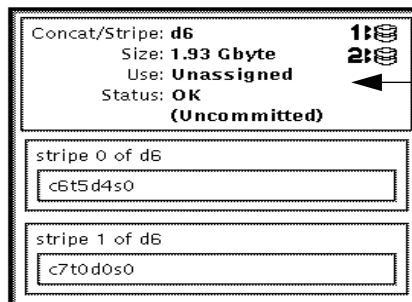
6. Select Concat as the type of device.

The Stripe or Concatenate dialog box is displayed. Click on the Concat button.



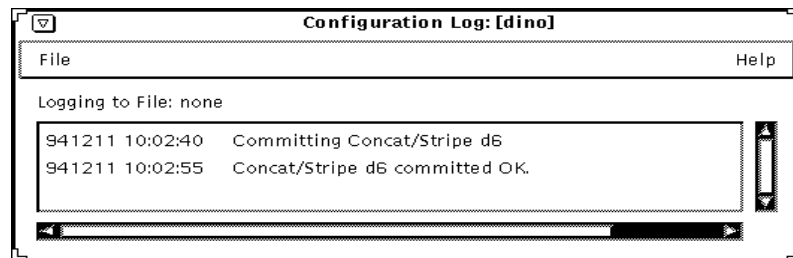
7. Commit the new device.

Click on the top rectangle of the Concat/Stripe object. Then click on the Commit button on the button panel. Information about the new devices is written to the appropriate files.



Click inside this rectangle.
Then click on the Commit button.

The Configuration Log shows Concat/Stripe d6 was committed.



8. Put the d6 Concat/Stripe object away.

Click on the top rectangle of the d6 Concat/Stripe object. Then click on the Put Away button on the button panel. The d6 Concat/Strip is placed in the device list.

Striped Metadevices

When components are configured into stripes, faster I/O throughput can be obtained. The ideal configuration of stripes would have each component on a separate disk controller. This configuration would deliver the best performance.

A configuration that delivers good performance would have the components on separate disks, but possibly on the same controller.

Note – Striping the slices on a single disk should be avoided. This will hurt performance.

You cannot change the interlace value of an existing stripe with DiskSuite Tool. To change the value, back up the metadevice, redefine the metadevice with a different value, and restore the data.

Figure 4-4 shows two physical components and a metadevice. The physical components are `/dev/dsk/c6t5d1s0` and `/dev/dsk/c7t0d1s0` and the metadevice is `/dev/md/dsk/d7`. The dotted rectangle represents the stripe defined in this example. The interlace value is 32 Kbytes.

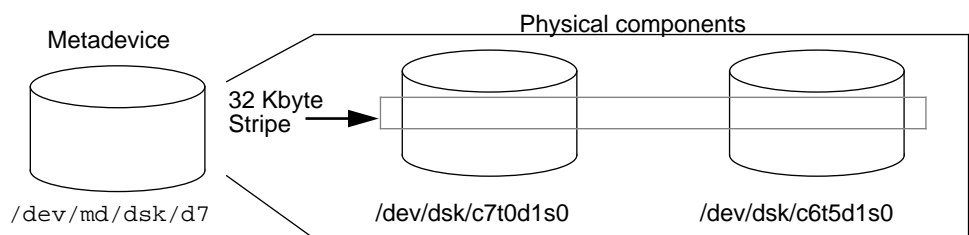


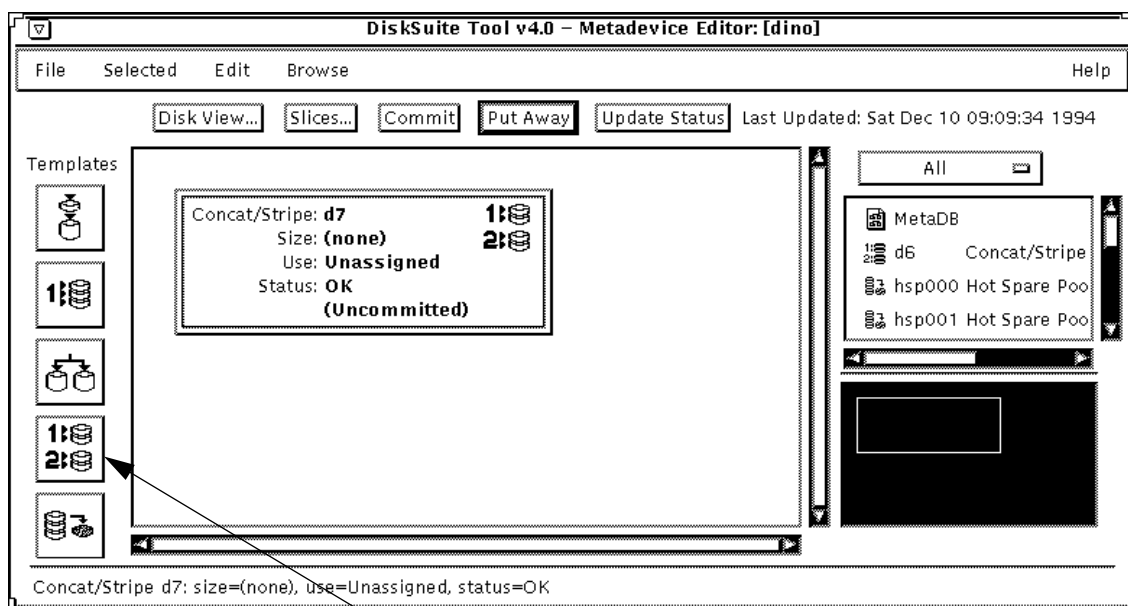
Figure 4-4 Striped Metadevice

To set up a stripe of the two physical components shown in Figure 4-4, by using DiskSuite Tool, perform the following steps:

1. Start DiskSuite Tool.

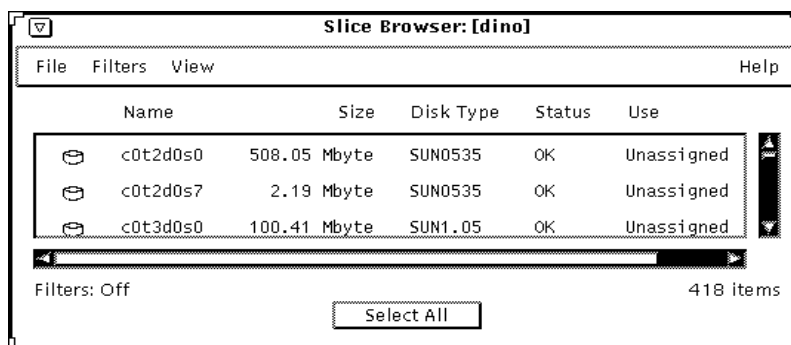
```
# metatool
```

2. **Open a Concat/Stripe template on the Metadevice Editor canvas.**
Click on the Concat/Stripe template. An unassigned and uncommitted Concat/Stripe object is opened on the Metadevice Editor canvas. When you point to the template, the message line shows the metadevice name, size, use, and status.



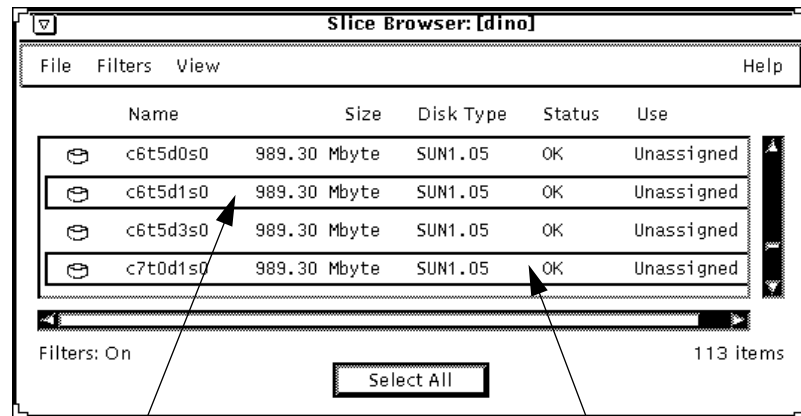
Click here to open the Concat/Stripe template

3. **Open the Slice Browser.**
Click on Slices on the button panel.



4. Select the slices that will be striped.

Scroll through the list of the devices on the Slice Browser and locate the two slices that will be striped. Click on the first slice that will be striped, then hold down the Control key and click on the second slice.



Click here to select the first slice.

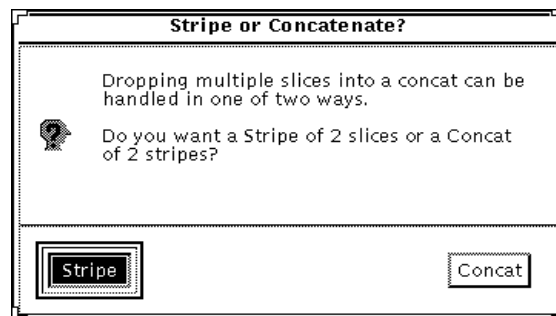
Hold down the Control key and click here to select the second slice.

5. Drag the two slices into the Concat/Stripe template.

Point to one of the two selected slices and hold down the middle button. Drag the slices to the Concat/Stripe d7 template and release the button.

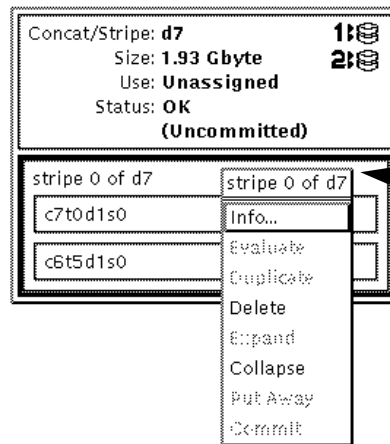
6. Select Stripe as the type of device.

The Stripe or Concatenate dialog box is displayed. Click on the Stripe button.



7. Optionally change the interlace size.

By default the interlace value of a stripe is 16 Kbytes. To change the interlace value, point to the rectangle where the “stripe 0 of d7” text appears, press the right button, and select the Info option.



Point inside this rectangle, press the right button, and select the Info option. The Stripe Information screen is displayed.

The Stripe Information window appears, displaying information about the stripe. To change the interlace value, click on the Custom button. Enter the new value in the fields. Click on the Attach button.

Stripe Information: [dino]

Device Name: **stripe 0 of d6**

Status: **OK**

Size: **600.47 Mbyte**

Interlace:

☐ Default 16 Kbytes
 ☒ Custom Kbytes

Attach

☒ Show Slices (2)

Slice	Size	Replicas	Status
c0t3d0s5	300.23 Mbyte	0	OK
c0t3d0s6	300.23 Mbyte	0	OK

Slice:

Default interlace value is set to 16 Kbytes.

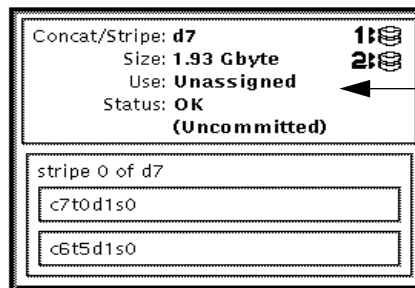
Click on Custom to change the interlace value.

The value entered is 32 Kbytes.

Point to the Attach button and click the left button to assign the new interlace value.

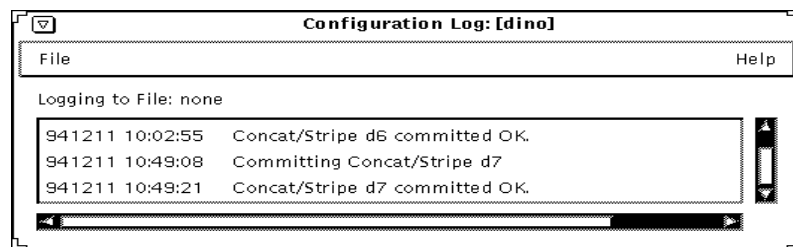
8. Commit the new device.

Click on the top rectangle of the Concat/Stripe object. Then click on the Commit button on the button panel. Information about the new devices is written to the appropriate files.



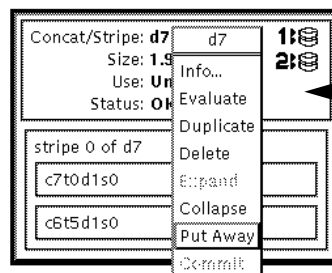
Click inside this rectangle.
Then click on the Commit button.

The Configuration Log shows Concat/Stripe d7 was committed.



9. Put the d7 stripe away.

Point inside the top rectangle of the Concat/Stripe object and press the right button. Select Put Away on the menu.



Point inside the Concat/Stripe
rectangle and press the right
button. Select the Put Away option.

Metadevices Defined as Concatenated Stripes

Concatenated stripes enable you to expand an existing stripe and provide some improved performance over simple concatenations.

Figure 4-5 shows three physical components and a metadevice. The physical components are `/dev/dsk/c6t5d3s0`, `/dev/dsk/c7t0d2s0`, and `/dev/dsk/c7t0d3s0`. The metadevice is `/dev/md/dsk/d8`. The first stripe (shown as a dotted rectangle) is made up of portions of two physical components and the second stripe is one component.

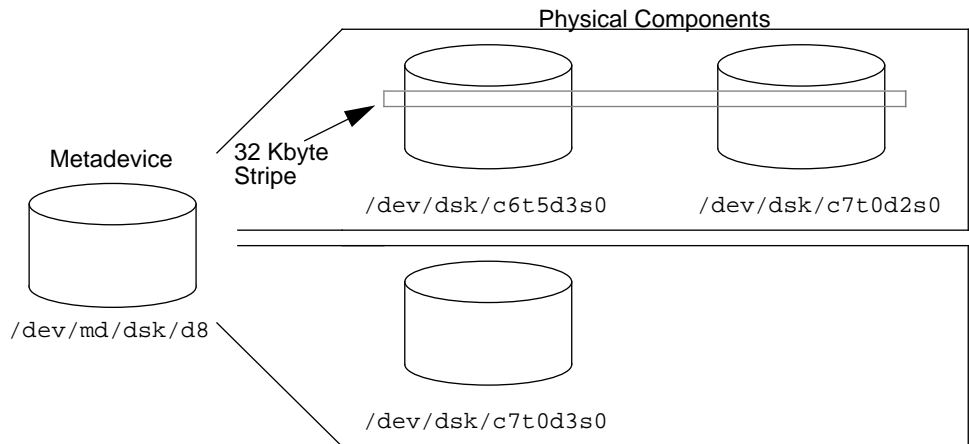


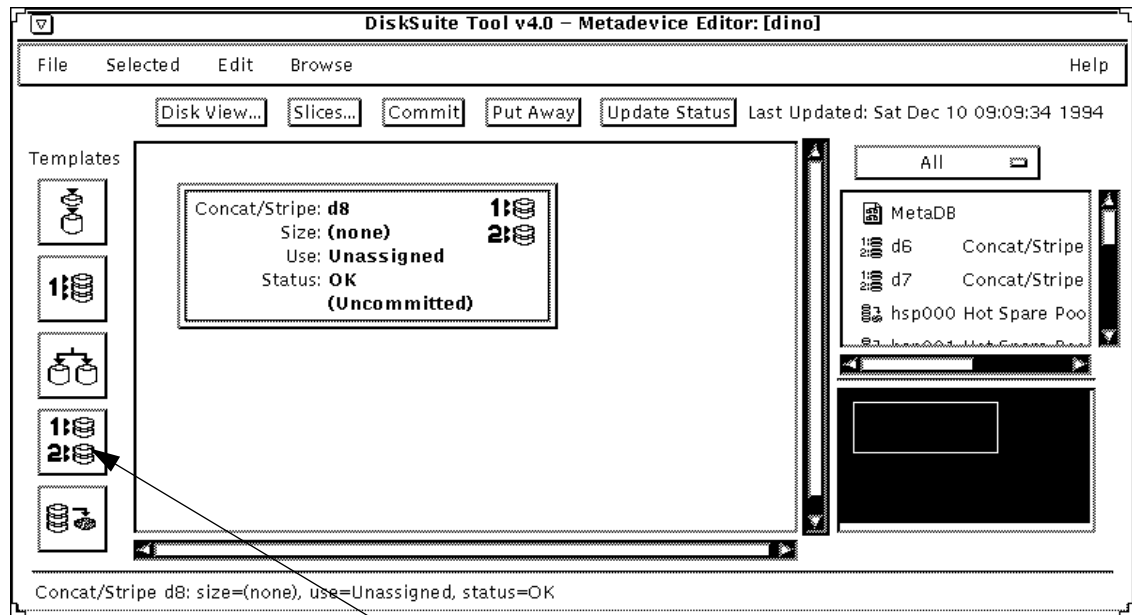
Figure 4-5 Concatenation of Two Stripes Into a Metadevice

To use DiskSuite Tool to set up the concatenated stripe shown in Figure 4-5, perform the following steps:

1. Start DiskSuite Tool.

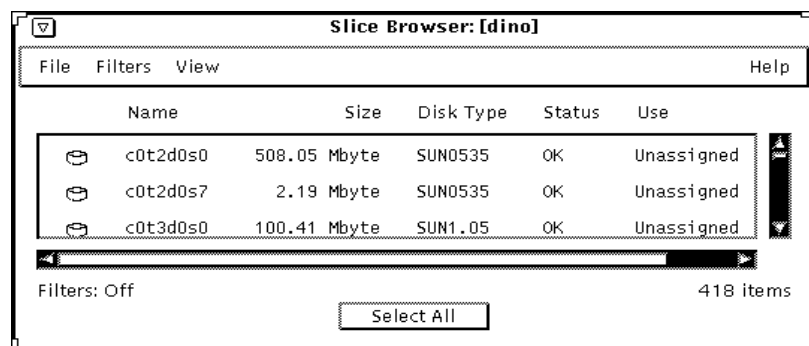
```
# metatool
```

2. **Open a Concat/Stripe template on the Metadevice Editor canvas.**
Click on the Concat/Stripe template. An unassigned and uncommitted Concat/Stripe template is opened. When you point to the template, the message line shows the metadevice name, size, use, and status.



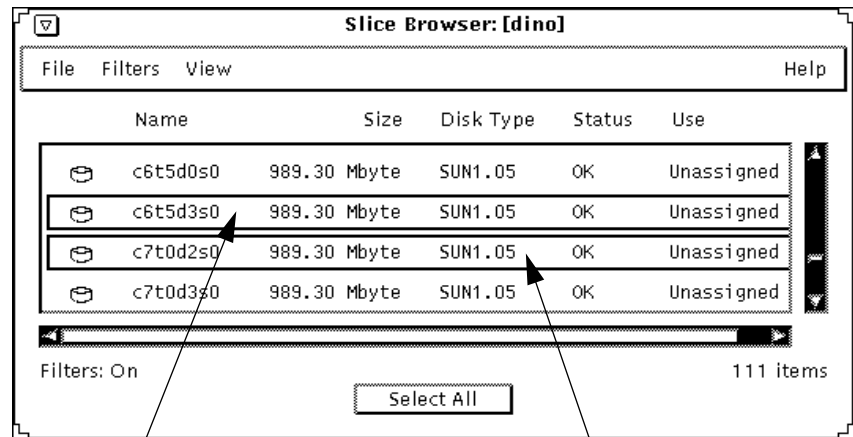
Click here to open the Concat/Stripe template

3. **Open the Slice Browser.**
Click on Slices on the button panel.



4. Select the two slices that will be striped.

Scroll through the list of the devices in the Slice Browser and locate the two slices that will be concatenated. Click on the first slice that will be concatenated, then hold down the Control key and click on the second slice.



Click here to select the first slice

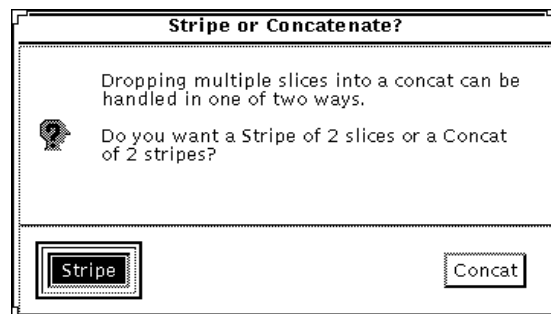
Hold down the Control key and click here to select the second slice.

5. Drag the two slices into the Concat/Stripe d8 template.

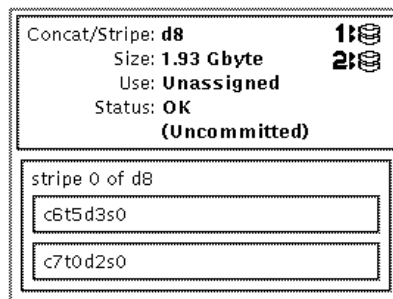
Point to one of the two selected slices and hold down the middle button. Drag the slices to the Concat/Stripe d8 template and release the button.

6. Select Stripe as the type of device.

The Stripe or Concatenate dialog box is displayed. Click on the Stripe button.

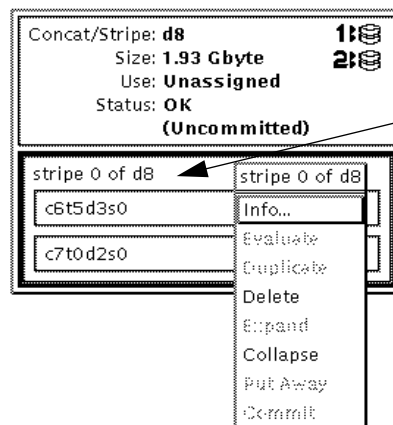


The stripe is populated with the two slices.

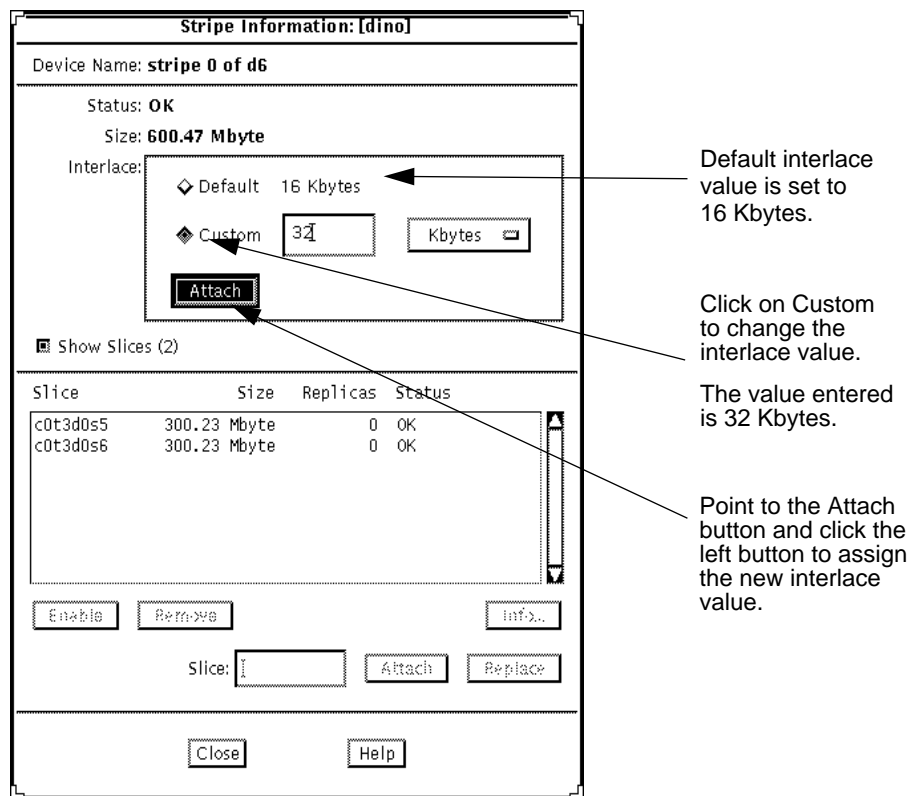


7. Optionally change the interlace size.

By default the interlace value of a stripe is 16 Kbytes. To change the interlace value, point inside the rectangle that contains the “stripe 0 of d8” text and press the right button. Select the Info option. The Stripe Information screen is displayed.



The Stripe Information window appears, displaying information about the stripe. To change the interlace value, click on the Custom button. Enter the new value in the fields. Click on the Attach button.



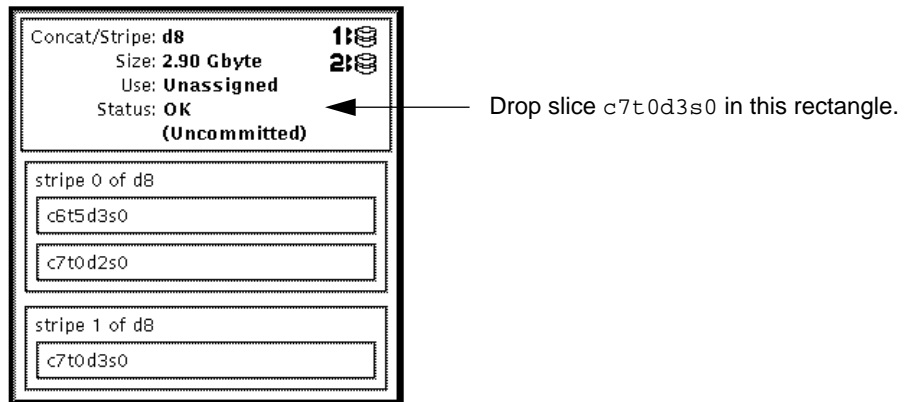
8. Select the slice to be concatenated to the stripe.

Scroll through the list of components in the Slice Browser to locate the slice that will be concatenated at the end of the d8 stripe (c7t0d3s0).

9. Drag the slice into the Concat/Stripe d8 template.

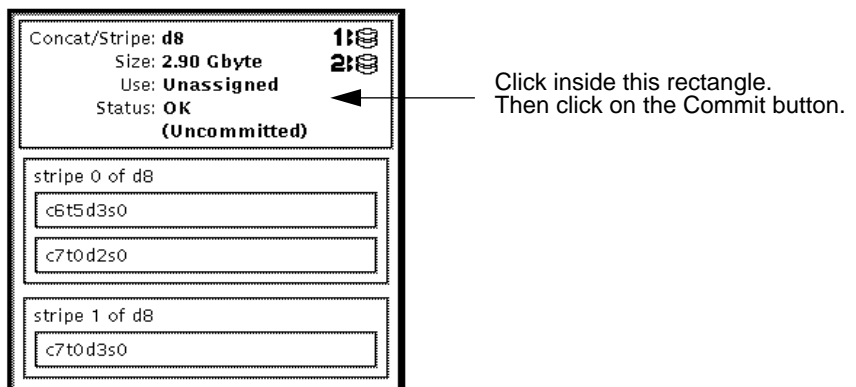
Point to the slice and press and hold down the middle button. Drag the slice to the top of the Concat/Stripe d8 template and release the button.

The concatenated stripe is then populated with the c7t0d3s0 slice.



10. Commit the d8 Concat/Stripe.

Click on the top rectangle of the d8 Concat/Stripe object. Then click on the Commit button on the button panel. Information about the new devices is written to the appropriate files.



11. Put the d8 Concat/Stripe away.

Click on the top rectangle of the Concat/Stripe object. Then click on the Put Away button on the button panel.

Replacing Failed Components

To remove a component from a concatenation or a stripe (for example, if a component is reporting soft errors), use one of the following methods:

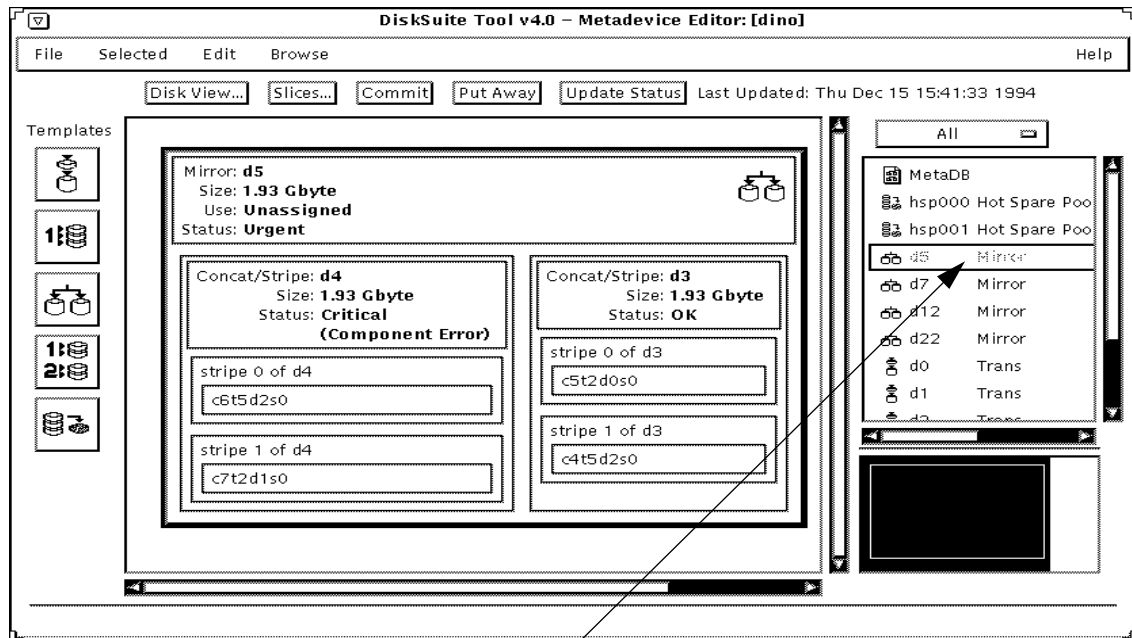
- If the metadvice is not in a mirror or in a RAID, use the instructions in Chapter 4 of the *Solstice DiskSuite 4.0 Administration Guide* to replace the failing component (if another disk is already attached to the system).
- If the metadvice is in a mirror (described in Chapter 5, “Mirroring”) or in a RAID (described in Chapter 9, “RAID Devices”), use DiskSuite Tool to replace the faulty component. The following example shows replacement of a failed component in a mirror.

1. Start DiskSuite Tool.

```
# metatool
```

2. Open the mirror on the Metadevice Editor canvas.

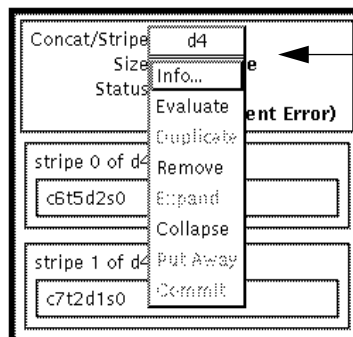
Double-click on the mirror with the error. In this example, mirror d5 has been opened. A component error with a slice in the d4 submirror is being reported. The status is Critical.



Double-click here to open the d5 Mirror.

3. Open the Slice Information window.

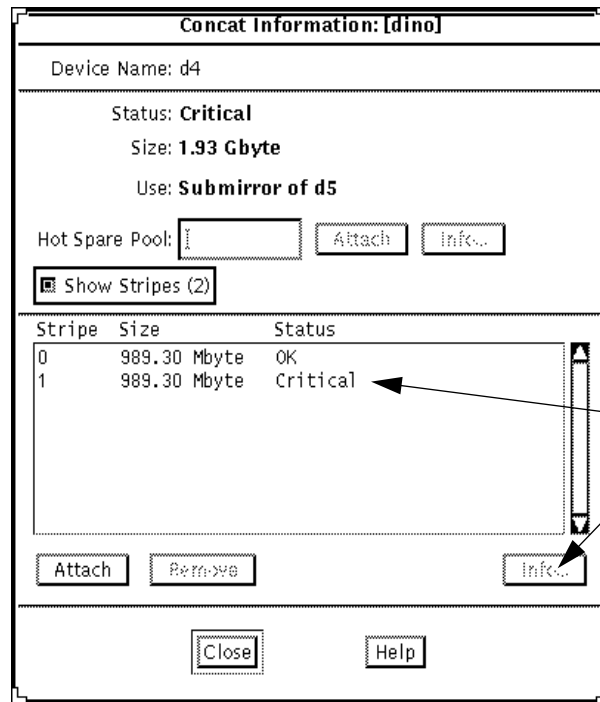
Point inside the submirror rectangle and press the right button. Select Info from the menu.



Point in the Concat/Stripe rectangle and press the right button. Select the Info option.

4. View the size and condition.

The Slice Information window displays the size and condition of the slice. The problem is with stripe 1, which is the slice `c7t2d1s0`. The replacement slice must be 989.30 Mbytes or larger.



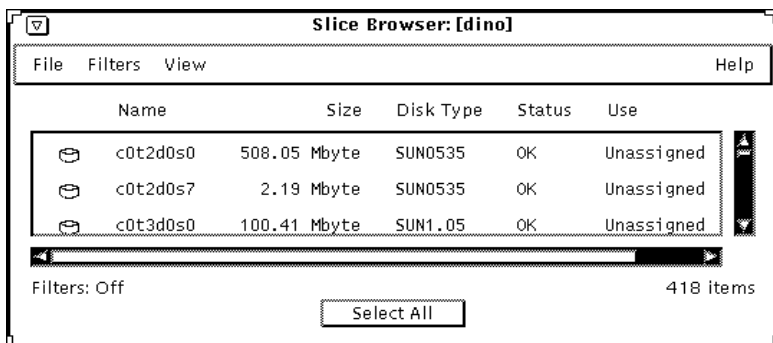
To obtain additional information about the slice with the error:

1. Click on the slice.

2. Click on Info.

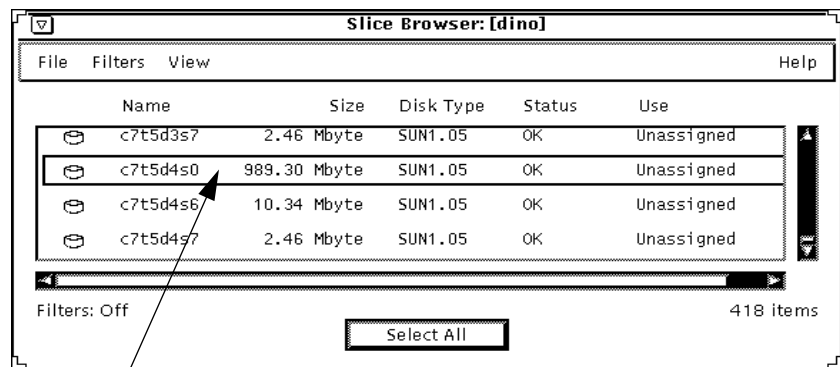
5. Open the Slice Browser.

Click on Slices on the button panel.



6. Locate a replacement slice.

Scroll through the list of slices and locate an unused slice that is 989.30 Mbytes or locate multiple slices that have a combined size of 989.30 Mbytes. The slice or slices selected should be attached to a different controller than the slices in the other submirror to help ensure availability of data. (Refer to “Setting Browser Filters” on page 50 for information about how to display slices of a certain size or other criteria.)

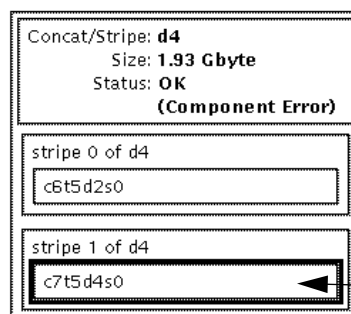


Click here to select the slice.

7. Drop the replacement slice on the errored component.

Point to the replacement slice. Press and hold down the middle button. Drag the cursor to the rectangle that contains the failed component, c7t2d1s0. Release the button.

Note – Be sure to point the cursor inside the rectangle that contains the device number of the errored component, instead of other rectangles inside the Concat/Stripe template.

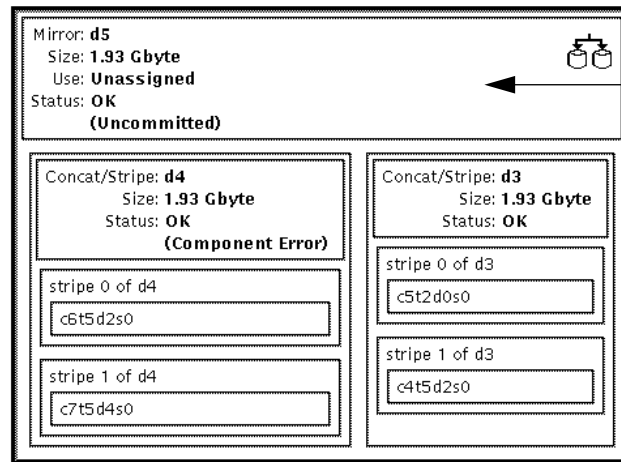


The replacement slice is dropped in the rectangle that contains the errored slice.

8. Commit the mirror.

Click on the top rectangle of the d5 Mirror. Then click on the Commit button.

Disk View... Slices... **Commit** Put Away Update Status Last Updated:



Click inside this rectangle. Then click on the Commit button.

Clearing Concatenations and Stripes

You can easily clear (delete) a concatenation or stripe and use the components in a traditional fashion.



Caution – Any data that is on the concatenation or stripes will be lost when the metadevice is cleared. Be sure to back up the data on the metadevice.

The procedure for clearing concatenations or stripes is described in the following example. In this example, the metadevice that is being cleared is named d7.

1. Back up the data.

Back up all data currently on the components that make up the metadevice (either the stripe or concatenation).

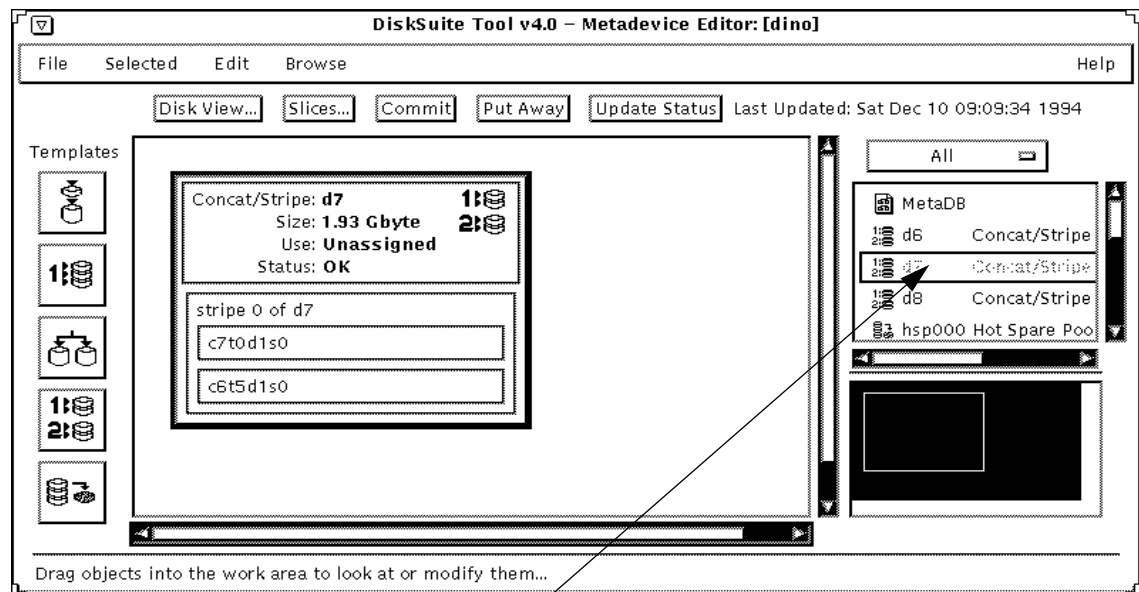
2. Stop access to the data on the metadvice.

For example, if a file system resides on the metadvice, unmount (with `umount (1M)`) the file system.

```
# /sbin/umount /dev/md/dsk/d7
```

3. Open the d7 Concat/Stripe on the Metadvice Editor canvas.

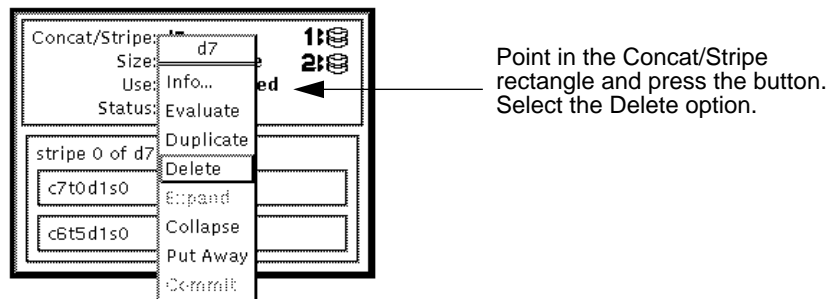
Point to the d7 Concat/Stripe in the device list and double-click the left button.



Double-click here to open the d7 Concat/Stripe.

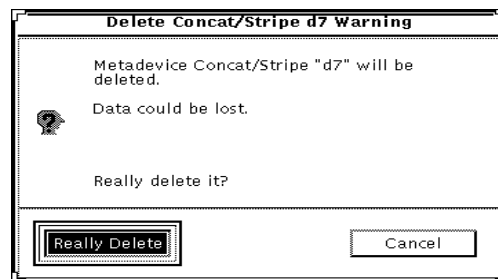
4. Bring up the menu and select the Delete choice.

Point inside the top of the d7 metadvice object and press the right button. Drag the cursor to the Delete choice and release the button.



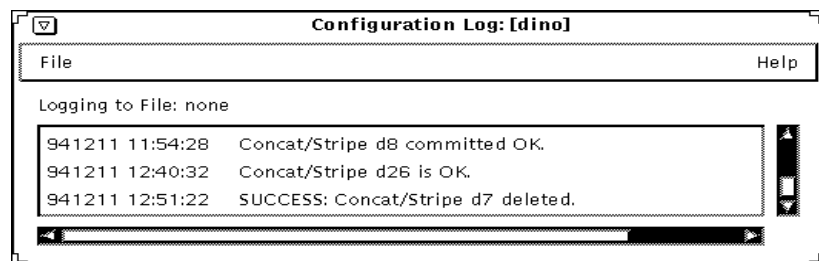
5. Select Really Delete when the dialog box is displayed.

Click on the Really Delete button.



6. Check the Configuration Log to ensure the metadvice was deleted.

When the deletion is completed, the Configuration Log window will pop to the front and display the message that the Concat/Stripe was deleted.



Note – Any hot spare pools assigned to d7 are preserved.

Concat Information Window

DiskSuite Tool supports several options to view and manipulate concatenations. The Concat Information window is shown in Figure 4-6. There are three ways to display the Concat Information window:

- Point to the Concat/Stripe object in the Metadevice Browser and double-click the left button. The Concat/Stripe object is opened on the Metadevice Editor's canvas and the Concat Information window will display.
- If the Concat/Stripe object is on the Metadevice Editor's canvas, point inside the template and bring up the concatenation menu. Select the Info choice.
- Double-click on a Concat/Stripe object on the Metadevice Editor's canvas.

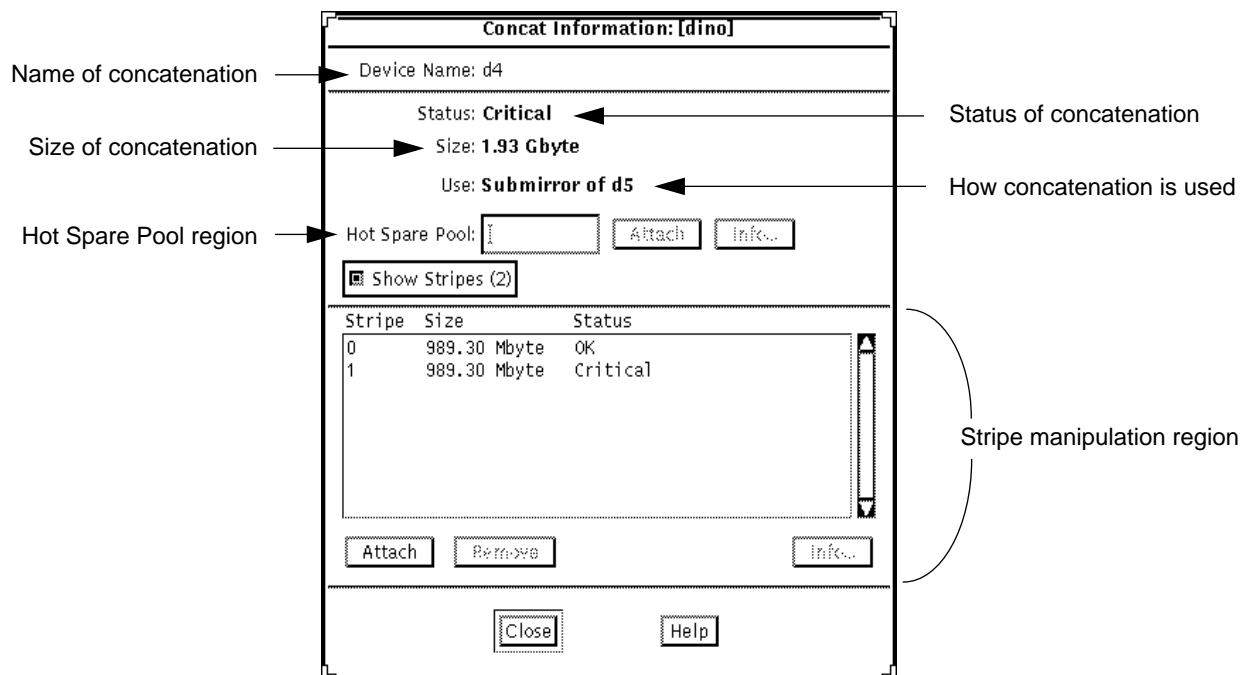


Figure 4-6 Concat Information Window

Table 4-1 lists the functionality associated with the regions of the Concat Information window.

Table 4-1 Concat Information Window Functionality

Field	Functions
Device name	The metadvice name of the concatenation.
Status	Description of the concatenation's status. See "Checking the Status of Concatenations and Stripes" on page 91" for a description of the values that appear in this region.
Size	The size of the concatenation.
Use	How the concatenation is currently used (for example, submirror).
Hot Spare Pool	The entry field for specifying the name of a Hot Spare Pool to be associated with the concatenation. To attach a hot spare pool enter the name in the field and click on the Attach button. The Hot Spare Pool Information window is displayed when you enter a hot spare pool name and click on the Info button.
Show stripes	This radio button enables you to turn on and off the stripe manipulation region. The number of stripes in the concatenation are shown in parentheses on the button.
Stripe manipulation region	The following functionality is available in this region: List of stripes – Provides the size and status of each stripe included in the concatenation. Attach – Attaches a new and empty stripe to the concatenation. Remove – Removes the selected stripe from the concatenation. Info – Brings up the Stripe Information window for the selected (highlighted) stripes.

Stripe Information Window

DiskSuite Tool supports several options for stripes. You set the options using choices on the Stripe Information window, as shown in Figure 4-7. There are three ways to display the Stripe Information window:

- Point inside the stripe rectangle of the Concat/Stripe object and double-click.
- If the Concat/Stripe object is on the Metadevice Editor's canvas, point inside the template and bring up the stripe menu. Select the Info choice.

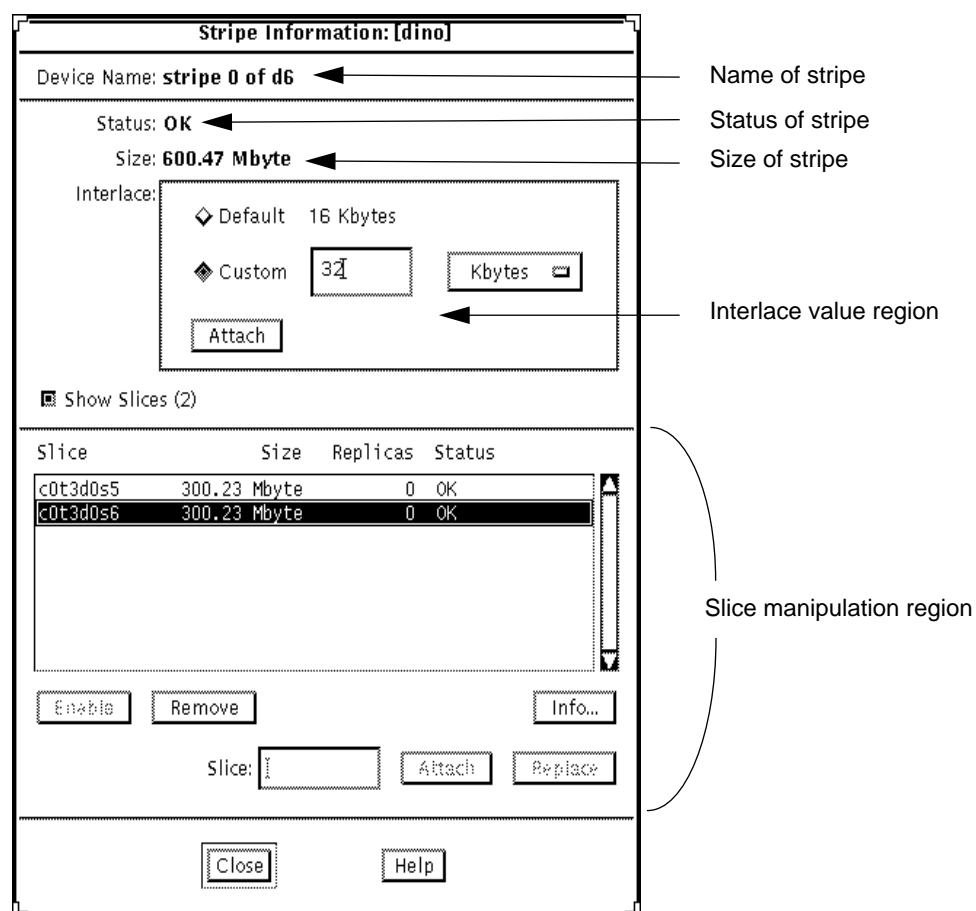


Figure 4-7 Stripe Information Window

Table 4-2 lists the functionality associated with the regions of the Stripe Information window.

Table 4-2 Slice Information Window Functionality

Field	Functions
Device name	The metadvice name of the stripe.
Status	Description of the stripe's status. See "Checking the Status of Concatenations and Stripes" on page 91" for a description of the values that appear in this region.
Size	The size of the stripe.
Interlace value region	The default interlace value is 16 Kbytes. To change the interlace value, click on the Custom button and enter the value in the field. The menu button to the right of the field enables you to specify the units used. The values on the menu are Gbytes, Mbytes, Kbytes, and Sectors. The default is Kbytes. After the Custom field is complete, the Attach button is used to assign the interlace value to the stripe. After a stripe is committed, the interlace value cannot be changed.
Show slices	This toggle button enables you to turn on and off the slice manipulation region. The number of slices in the stripe are shown in parentheses on the button.
Slice manipulation region	The following functionality is available in this region: Scrolling List – Shows slices included in the stripe. The information in this region includes the name of the slice, size, number of state database replicas on the slice, and the status. Enable – Activates the selected slices if they are disabled. Remove – Removes the selected slices. Slice – Enables you to specify a new slice to attach to the stripe or replace the selected slice. If no slice is selected, the button is unavailable. Attach – Attaches the slice specified in the Slice field to the stripe. This button is active only when a slice name is entered in the field. Replace – Replaces the selected stripe with the slice entered in the Slice field. This button is active only when a slice name has been entered in the field and a slice is selected on the scrolling list. Info – Displays the Slice Information window for the selected (highlighted) slice.

Checking the Status of Concatenations and Stripes

There are three ways to check the status of a Concat/Stripe metadvice using DiskSuite Tool. The methods are:

- Move the Concat/Stripe metadvice from the device list and check the Status field on the object.
- Bring up the Concat/Stripe Information window and check the Status field.
- Bring up the Problem List, which is available using the Browse menu on the menu bar of the Metadvice Editor window. If a problem exists, a detailed description of the problem is given.

The Status fields of the Concat/Stripe metadvice use the status keywords explained in Table 4-3.

Table 4-3 Concat/Stripe Status Keywords

Keyword	Meaning
OK	The Concat/Stripe metadvice has no errors and is functioning correctly.
Attention	There is a problem with the Concat/Stripe metadvice, but there is no immediate danger of losing data.
Urgent	There is a problem with the Concat/Stripe metadvice and you are only one failure away from losing data.
Critical	Data has potentially been corrupted.

Slices may be reported with any of the additional keywords shown in Table 4-4.

Table 4-4 Stripe Status Keywords

Keyword	Meaning
OK	The slice is part of a metadvice that reports its status as okay.
Enabled	The slice is currently out of service but is scheduled to be enabled the next time you click on the Commit button.
Resyncing	The slice is currently being resynced. This happens after you click on Enable or Remove on the Slice Information window.
Spared	The slice is currently being replaced in the metadvice by a slice from a Hot Spare Pool.

Mirroring



This chapter provides information and procedures for mirroring components using the Solstice DiskSuite Tool (`metatool(1M)`). Refer to Chapter 3, “Overview of DiskSuite Tool,” for information on using the DiskSuite Tool.

Use the following table to locate information specific to your task.

<i>Overview of Mirroring</i>	<i>page 94</i>
<i>Operation of Mirrors</i>	<i>page 94</i>
<i>Creating Mirrors</i>	<i>page 95</i>
<i>Mirroring Existing File Systems</i>	<i>page 100</i>
<i>Unmirroring File Systems</i>	<i>page 108</i>
<i>Mirroring File Systems You Cannot Unmount</i>	<i>page 110</i>
<i>Unmirroring File Systems You Cannot Unmount</i>	<i>page 120</i>
<i>Recording and Booting From Alternate Root File Systems</i>	<i>page 120</i>
<i>Reconfiguring Submirrors</i>	<i>page 121</i>
<i>Using Mirrors for Online Backup</i>	<i>page 137</i>
<i>Mirror Information Window</i>	<i>page 137</i>
<i>Checking the Status of Mirrors</i>	<i>page 141</i>

Overview of Mirroring

DiskSuite provides the ability to replicate data stored on a particular metadvice onto as many as two additional (for a total of three) metadvice, referred to as submirrors. By setting up a minimum configuration of a two-way mirror you can recover from a single-component failure and perform online backups of file systems.

Mirroring components improves data reliability. An error on a component does not cause the entire mirror to fail. For continued reliability after failure, DiskSuite provides a facility for creating hot spare components. For further information on this utility, see Chapter 7, “Hot Spares.”

Although mirroring helps ensure data reliability, it can cause the I/O performance to suffer on some systems. However, the performance penalty can be minimized by the correct configuration of the mirrors.

To get maximum protection and performance, place mirrored metadvice on different physical components (disks) and on different disk controllers. Since the primary purpose of mirroring is to maintain availability of data, defining mirrored metadvice on the same disk is not recommended. If the disk were to fail, both metadvice would fail.

Operation of Mirrors

Mirroring is accomplished by the creation of mirrors. A mirror is a special type of metadvice that is made up of one or more other metadvice. Each metadvice within a mirror is called a submirror.

Mirrors use the same naming convention as other metadvice (concatenations and stripes). After a mirror is defined, additional submirrors can be added at any time without disruption of writes or reads to the existing mirror.

Mirrors should be created with a single submirror at first. Then additional submirrors can be added. When you are mirroring existing file systems or data, be sure the existing data is contained on the submirror initially defined with the mirror. When a second submirror is subsequently attached, data from the initial submirror is copied onto the attached submirror automatically.

If you are creating mirrors from slices that contain existing data, you must create a one-way mirror of the slices where the data exists, then attach the second submirror.



Caution – In cases where you are mirroring existing data, be sure that the initial submirror contains the data you want mirrored.

Creating Mirrors

DiskSuite Tool provides two methods for creating mirrors:

- Define a one-way mirror using the templates. This is accomplished by dragging and dropping slices into a Concat/Stripe template to create either a concatenation or stripe. You then drag the newly created Concat/Stripe object into a mirror template and select Commit.
- Define a one-way mirror using the Mirror Information window, which is explained in “Mirror Information Window” on page 137. This is done by first filling in the names of the devices for one submirror, then entering the names of the other devices.

The first method is used throughout this section.

Mirroring Unused Slices

The following example shows the creation of metadevices (concatenations) and one-way mirrors using the drag-and-drop functionality of DiskSuite Tool.

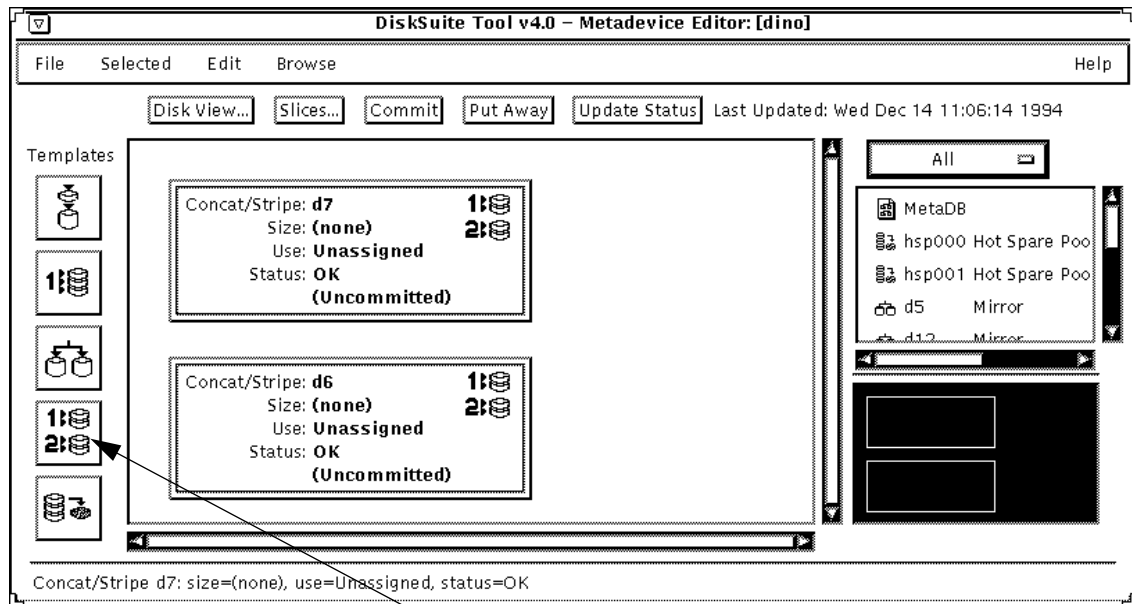
The example is a mirror that is made up of two submirrors, each containing two physical components that make up the metadevices. The physical components are named `c4t0d1s0`, `c5t0d1s0`, `c6t1d2s0`, and `c7t0d3s0`. The two submirrors are named `d6` and `d7`. The Mirror is named `d8`.

To set up this mirror of unused slices using the DiskSuite Tool, you would perform the following steps:

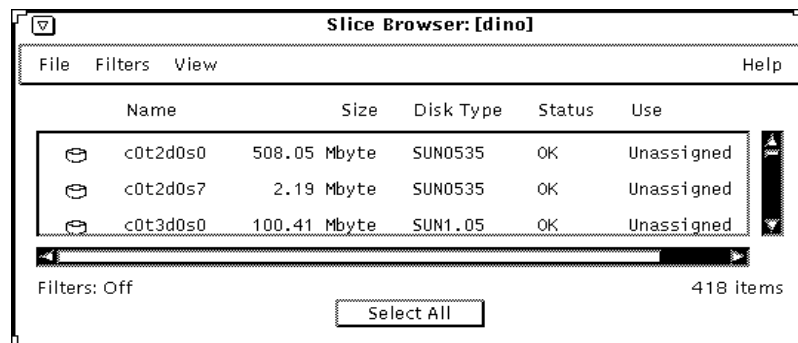
1. Start DiskSuite Tool.

```
# metatool
```

2. **Open two Concat/Stripe templates on the Metadevice Editor canvas.**
Click on the Concat/Stripe template twice. The Concat/Stripe templates opened are d6 and d7. Each of the two metadevices created in the Concat/Stripe templates will be a submirror in the mirror you are creating.



3. **Open the Slice Browser.**
Click on Slices on the button panel.



4. Select slices for the Concat/Stripe templates.

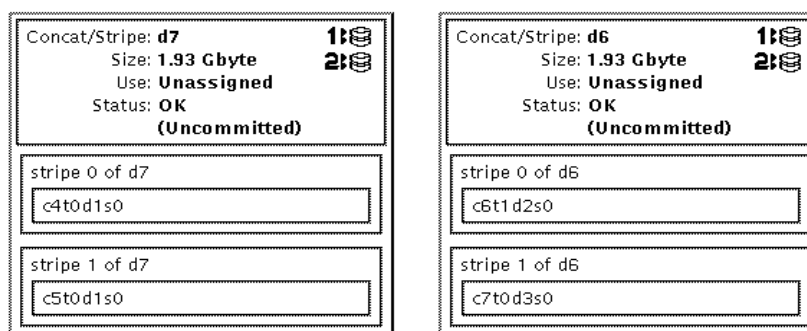
Scroll through the list of the devices in the Slice Browser and locate the slices that will be placed in the Concat/Stripe templates. The slices you select for one concatenation or stripe should be from disks that are attached to different controllers than those selected for the other concatenation or stripe. This will help ensure availability of the data if a controller fails. Also, approximately the same size slices should be selected for each of the Concat/Stripe templates. Any mismatch in the submirror sizes will lessen capacity, as the mirror will be only as large as the smallest submirror.

Click on the first slice that will be placed in the Concat/Stripe object, then hold down the Control key and click on the second. (Multiple slices can also be selected by holding down the left button and dragging the mouse cursor.)

5. Drag the slices into the Concat/Stripe templates.

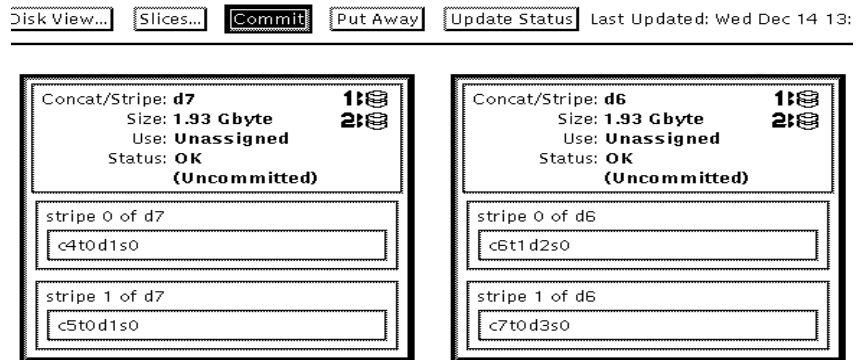
Point to one of the selected slices, press and hold down the middle button, and drag the slice to the Concat/Stripe template. Repeat the procedure for the second Concat/Stripe template.

In this example, each of the Concat/Stripe templates have been populated with two 989.30-Mbyte slices. The slices were concatenated to create two 1.93-Gbyte metadevices. (Refer to Chapter 4, “Concatenating and Striping,” for additional information on creating concatenations and stripes.)



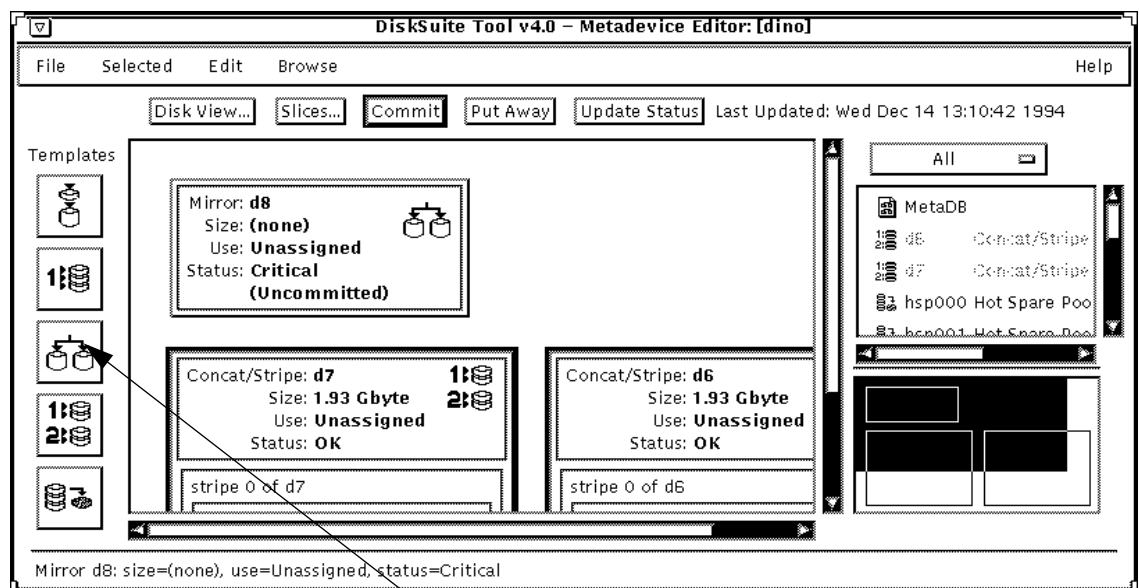
6. Commit the two concatenations.

Click on the top rectangle of one of the Concat/Stripe objects. Hold down the Control key and click on the top rectangle of the second. Then click on the Commit button.



7. Open a Mirror template on the Metadevice Editor Canvas.

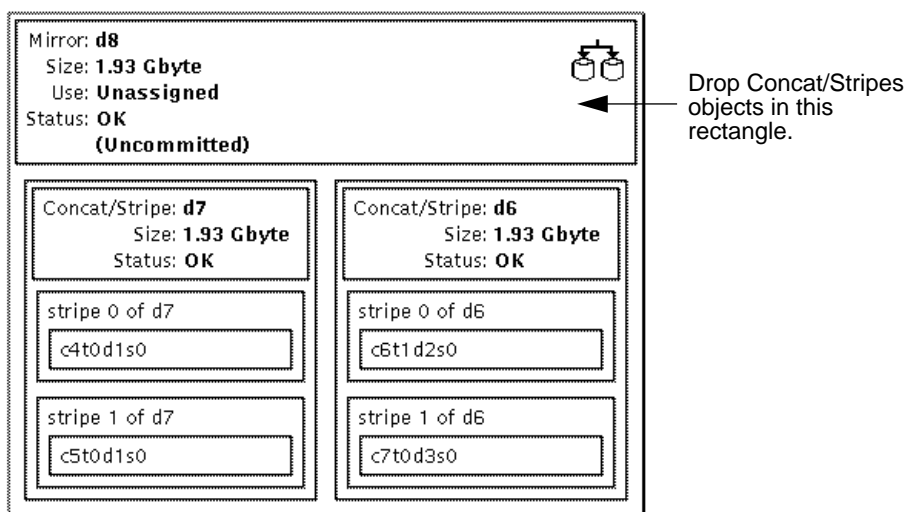
Click on the Mirror template. An unassigned Mirror template is opened. When you point to the template, the message line displays the metadevice name, size, use, and status.



Click here to open the Mirror template.

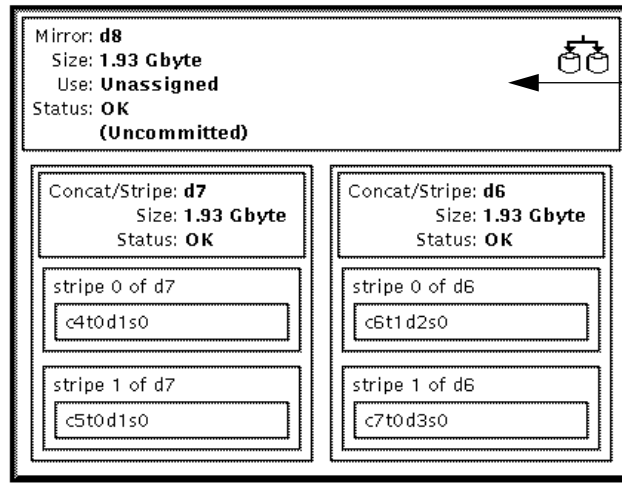
8. Drag the Concat/Stripe templates into the Mirror template.

Click on the top rectangle of one of the Concat/Stripe objects. Hold down the Control key and click on the top rectangle of the second. Point inside one of the Concat/Stripe templates and press and hold down the middle button. Drag the objects into the Mirror template.



9. Commit the Mirror.

Click on the top rectangle of the Mirror. Then click on the Commit button.



Click inside this rectangle. Then click on the Commit button.

Mirroring Existing File Systems

DiskSuite Tool enables you to mirror existing file systems without backing up data or reconfiguring devices.

Note – Because the root (/), swap, and some other file systems contain binary files that are necessary when the system is booting, special instructions must be used to mirror these file systems. Refer to “Mirroring File Systems You Cannot Unmount” in this chapter for detailed instructions.

To mirror an existing file system, you must use an additional component of equal or greater size. It is possible to use a concatenation of two or more components that have adequate space available to contain the submirror. For example, if /var is mounted on a 1-Gbyte component named c4t0d2s0, at least one Gbyte of space must be available on the metadvice being defined for the new submirror.

When you mirror an existing file system, first configure a one-way mirror with a submirror containing the existing file system. A submirror that you add subsequently should not contain any data that is needed, because it will be overwritten by the resync that follows the attachment.

The following example shows the step-by-step creation of a metadvice (Concat/Stripe), which is then added to an existing one-way mirror to create a two-way mirror. This is done using the drag and drop functionality of DiskSuite Tool.



Caution – The instructions for mirroring an existing file system are very similar to the instructions for mirroring unused slices. However, it is important to follow the procedure in the correct order or the mirroring will not function properly and existing data will be lost.

The example is a mirror that is made up of two physical components and two submirrors (metadevice). The physical components are named `c4t0d2s0` and `c5t0d2s0`. The two submirrors are named `d9` and `d17`. The Mirror is named `d18`.

To set up this mirror of an existing file system using DiskSuite Tool, you would perform the following steps:

1. Unmount the file system.

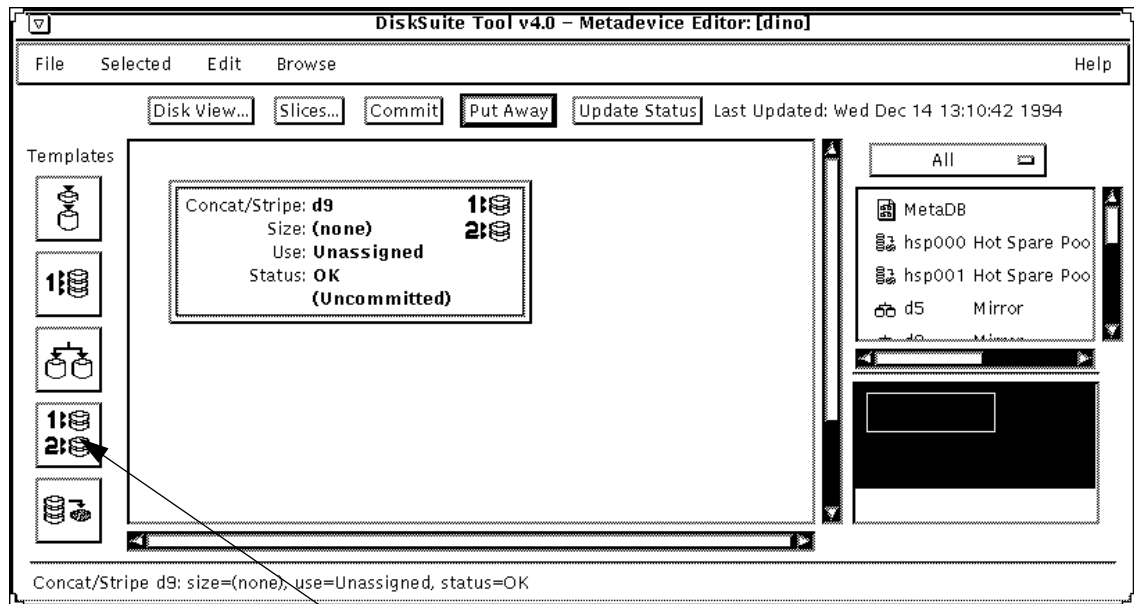
In this example, the existing file system is named `/var` and is on slice `c4t0d2s0`. If the file system is busy, an error message is displayed.

```
# /sbin/umount /var
```

2. Start DiskSuite Tool.

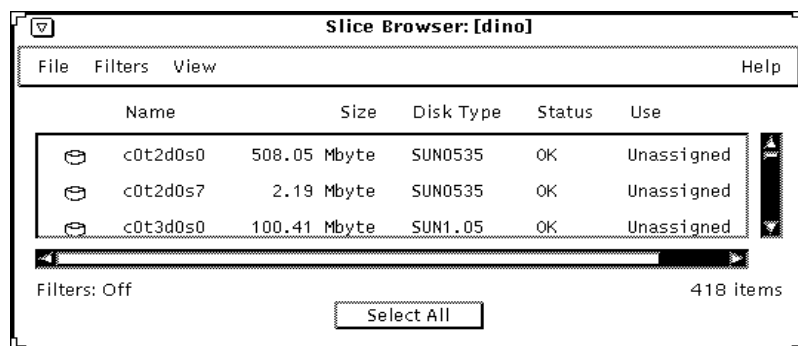
```
# metatool
```

3. **Open a Concat/Stripe template on the Metadevice Editor canvas.**
Click on the Concat/Stripe template. An unassigned and uncommitted Concat/Stripe template is opened. When you point to the template, the message line displays the metadevice name, size, use, and status.



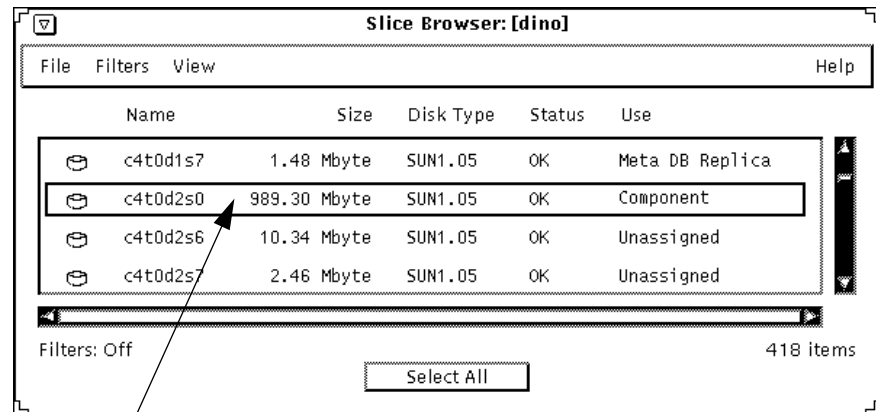
Click here to open a Concat/Stripe template.

4. **Open the Slice Browser.**
Click on Slices on the button panel.



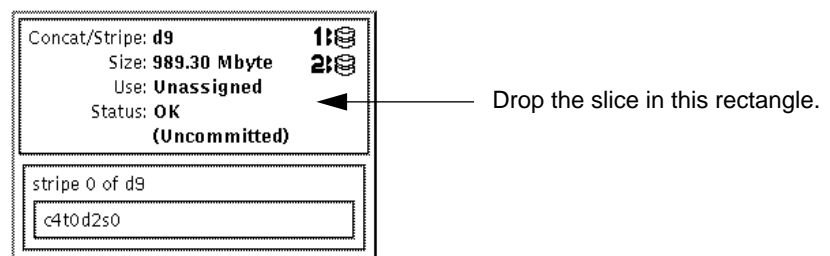
5. Locate the slice where `/var` resides.

Click on slice `c4t0d2s0` in the Slice Browser. Note that the Use field does not reflect that the slice is where the `/var` resides because the file system has been unmounted.



6. Drop the slice into the Concat/Stripe template.

Point to slice `c4t0d2s0`, press the middle button, and drag the slice to the Concat/Stripe `d9` template.



7. Open another Concat/Stripe template.

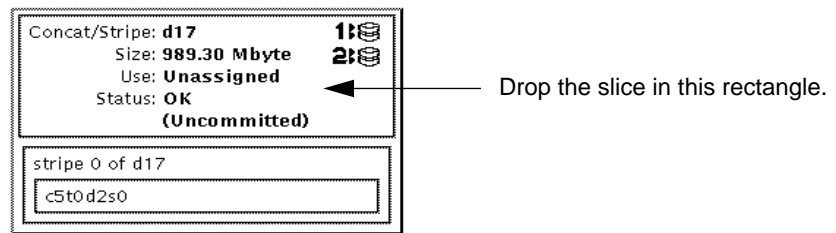
Click on the Concat/Stripe template.

8. Select another slice.

Scroll through the list of slices in the Slice Browser and locate a slice that is the same size or larger than c4t0d2s0. To help ensure availability, select a slice that is attached to a different controller than c4.

9. Drop the slice in the Concat/Stripe template.

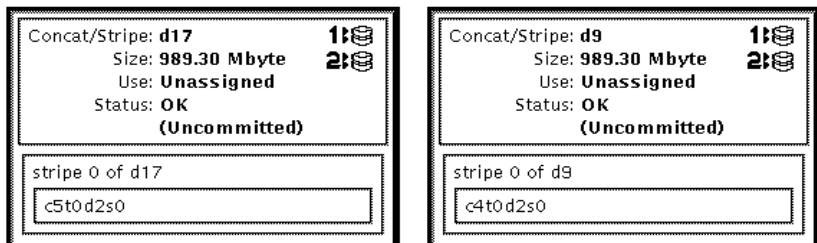
Point to slice c5t0d2s0 and hold down the middle button. Drag the slice to the Concat/Stripe d17 template.



10. Commit the two Concat/Stripe templates.

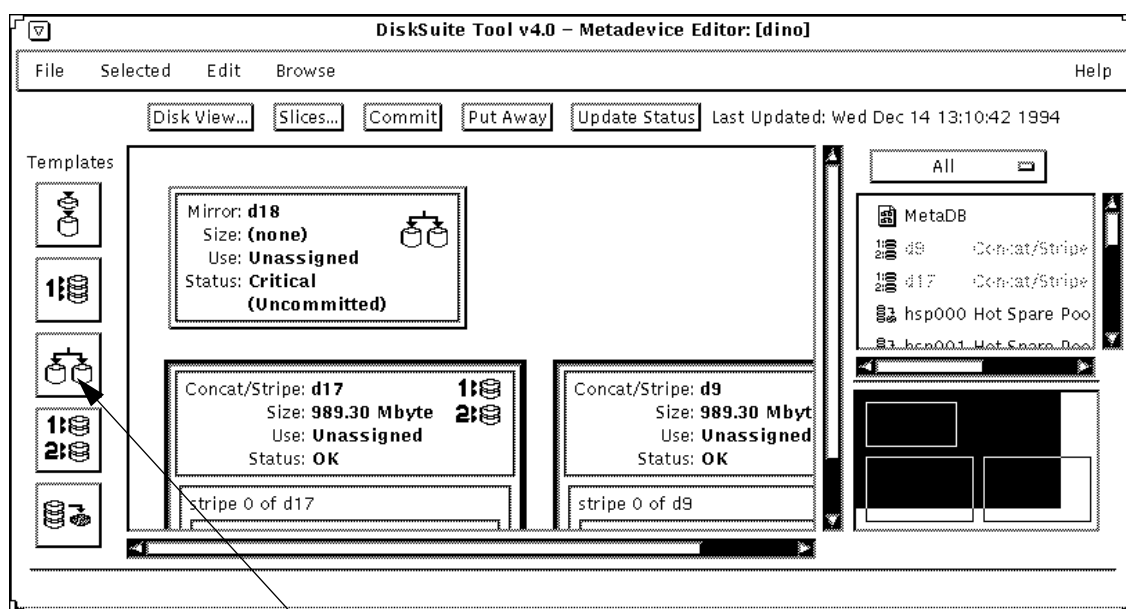
Click on the top rectangle of one of the Concat/Stripe objects. Hold down the Control key and click on the top rectangle of the second. Then click on the Commit button.

Disk View... Slices... **Commit** Put Away Update Status Last Updated: Wed Dec 14 13



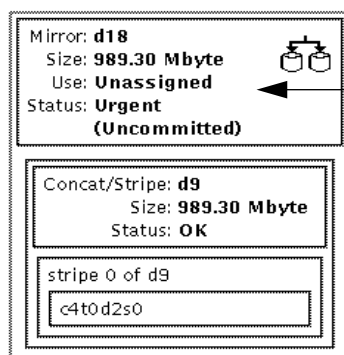
11. Open a Mirror template on the Metadevice Editor Canvas.

Click on the Mirror template. An unassigned Mirror template is opened. When you point to the template, the message line displays the metadevice name, size, use, and status.



12. Drag the d9 Concat/Stripe into the Mirror template.

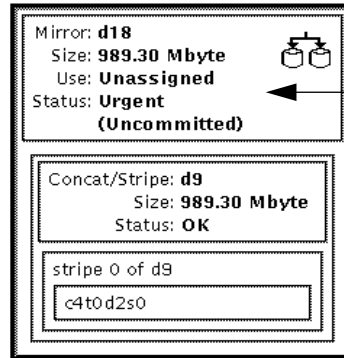
Drag the d9 Concat/Stripe, which contains the slice where the /var file system resides, into the Mirror template. The d9 Concat/Stripe object must be the first one placed in the Mirror template.



Drop the Concat/Stripe in this rectangle.

13. Commit the Mirror.

Click on the top rectangle of the Mirror. Then click on the Commit button.



Click inside this rectangle.
Then click on the Commit button.

14. Edit the /etc/vfstab file.

After you complete the previous steps, you must change the entry for /var in the /etc/vfstab file. For example, the following line:

```
/dev/dsk/c4t0d2s0 /dev/rdisk/c4t0d2s0 /var ufs 4 yes -
```

should be changed to read:

```
/dev/md/dsk/d18 /dev/md/rdisk/d18 /var ufs 4 yes -
```

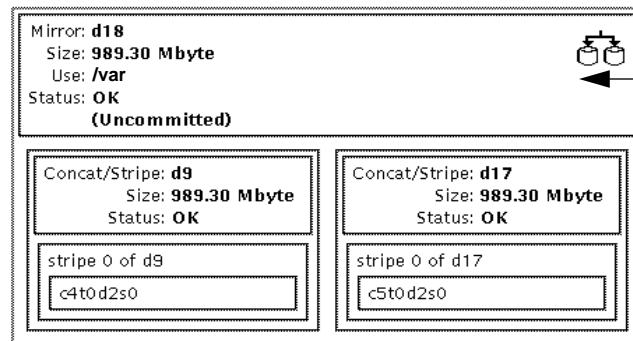
15. Mount the /var file system.

Enter the following command to mount the /var file system.

```
# /sbin/mount /var
```

16. Drag the other Concat/Stripe into the Mirror template.

Point to the d17 Concat/Stripe template and hold down the middle button. Drag the object into the Mirror template. A resync of the Mirror (copying all the information from d9 to d17) automatically occurs. The Use field of the Mirror will change to `/var` after you press the Update Status button in the button panel.

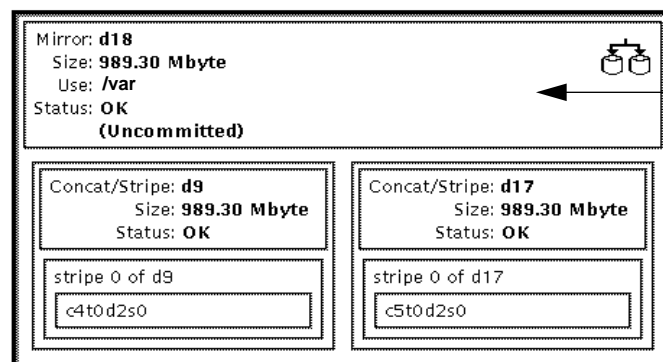


Drop Concat/Stripe objects in this rectangle.

17. Commit the Mirror again.

Click on the top rectangle of the d18 Mirror. Then click on the Commit button.

[Disk View...](#)
[Slices...](#)
[Commit](#)
[Put Away](#)
[Update Status](#)
[Last Update](#)



Click inside this rectangle. Then click on the Commit button.

Unmirroring File Systems

This subsection contains the procedure for unmirroring file systems that can be unmounted while the system is running. The following is an example of unmirroring the `/var` file system. In this example, `/var` is made up of a two-way Mirror with the names `d9` and `d17` in a Mirror named, `d18`. The names of the components are `c4t0d2s0` and `c5t0d2s0`.

The steps to follow when unmirroring the file system are:

1. Unmount the `/var` file system.

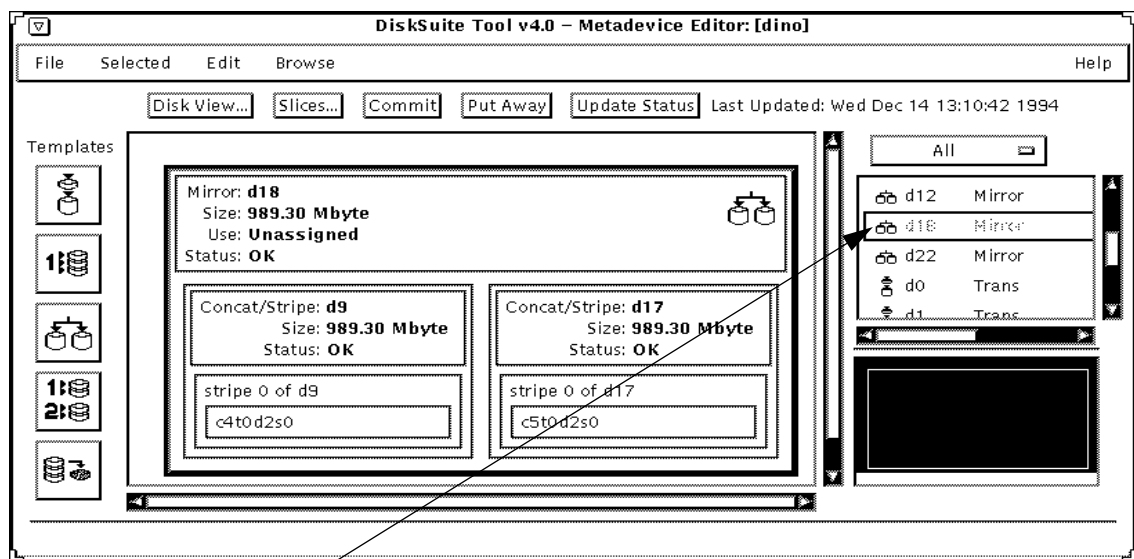
```
# /sbin/umount /var
```

2. Start DiskSuite Tool.

```
# metatool
```

3. Open the `d18` Mirror.

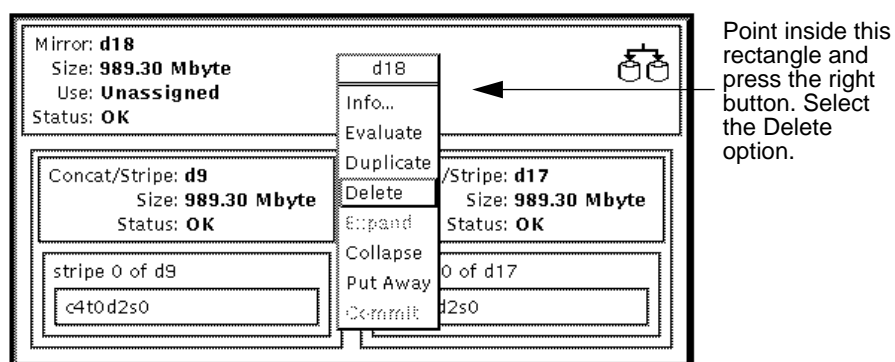
Scroll through the device list in the Metadevice Editor window and locate the `d18` Mirror. Point to the Mirror and double-click.



Double-click here to open the Mirror.

4. Delete the Mirror.

Point inside the top rectangle of the Mirror and press the right button to bring up the menu. Select the Delete option.



5. Confirm the deletion of the Mirror.

A Confirmation dialog box asks if you are sure you want to delete the Mirror. Click on the Really Delete button. This dialog box appears even if all the steps are followed correctly.



6. Edit the /etc/vfstab file.

After you complete the previous steps, you must change the entry for /var in the /etc/vfstab file. For example, the following line:

```
/dev/md/dsk/d18 /dev/md/rdisk/d18 /var ufs 4 yes -
```

should be changed to read:

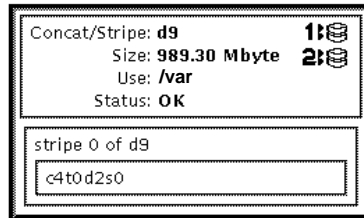
```
/dev/md/dsk/d9 /dev/md/rdisk/d9 /var ufs 4 yes -
```

7. Remount the /var file system.

Enter the following command to mount the /var file system.

```
# /sbin/mount /var
```

When /var is remounted, the Concat/Stripe d9 contains the file system.



To have the file system mounted on the slice rather than in the Concat/Stripe, you must first delete the Concat/Stripe. Then edit /etc/vfstab file to read:

```
/dev/dsk/c4t0d2s0 /dev/rdisk/c4t0d2s0 /var ufs 4 yes -
```

Mirroring File Systems You Cannot Unmount

DiskSuite Tool supports mirroring the root (/) and swap file systems, along with any other file systems (for example, /opt and /usr) that cannot be unmounted while the system is running.

Note – If problems occur when mirroring root (/), swap, and /usr, refer to Appendix C of the *Solstice DiskSuite 4.0 Administration Guide*. This appendix explains how to recover from a failed boot.



Caution – When you are mirroring the root file system, you must record the secondary root slice name so you can reboot the system if the primary submirror fails. This information should be written down, not recorded on the system which may not be available. Read the section, “Recording and Booting From Alternate Root File Systems” on page 120 for details.

Note – DiskSuite Tool does not support mirroring root (/) with submirrors that are made up of more than one slice, because the system cannot boot.

This subsection contains an example that places the root (/) file system on a two-way mirror. The same instructions apply to mirroring `swap` and other file systems that cannot be unmounted (for example, `/opt` or `/usr`).

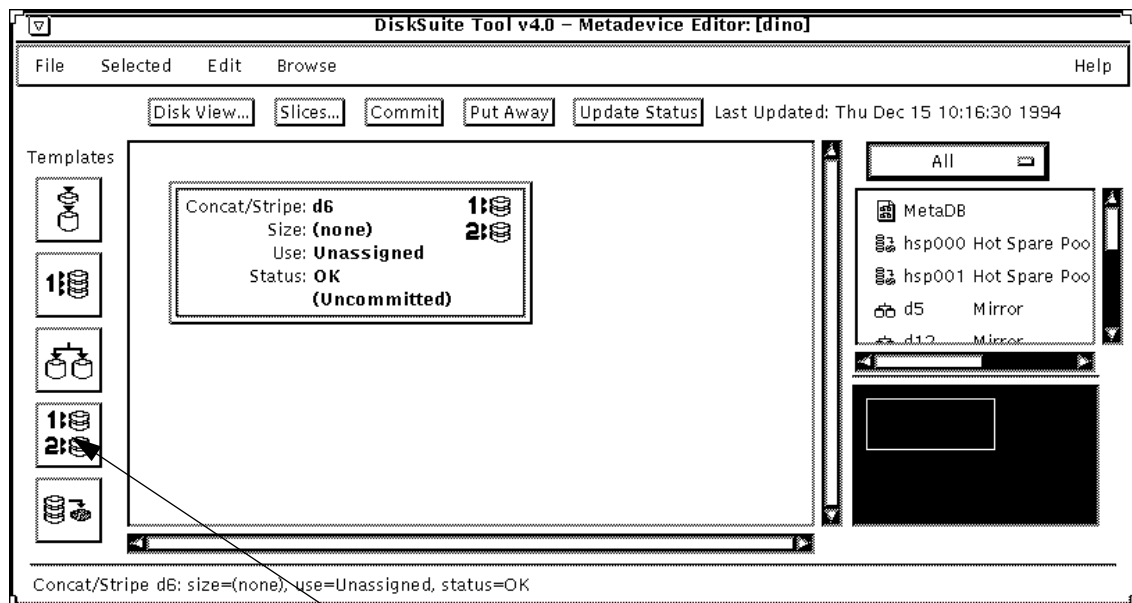
In this example, root (/) resides on `c0t2d0s0`. It will be mirrored on `c0t3d0s0`. The names of the Concat/Stripe objects will be `d6` and `d8`. The name of the Mirror will be `d7`. The steps to follow are:

1. Start DiskSuite Tool.

```
# metatool
```

2. Open a Concat/Stripe template on the Metadevice Editor canvas.

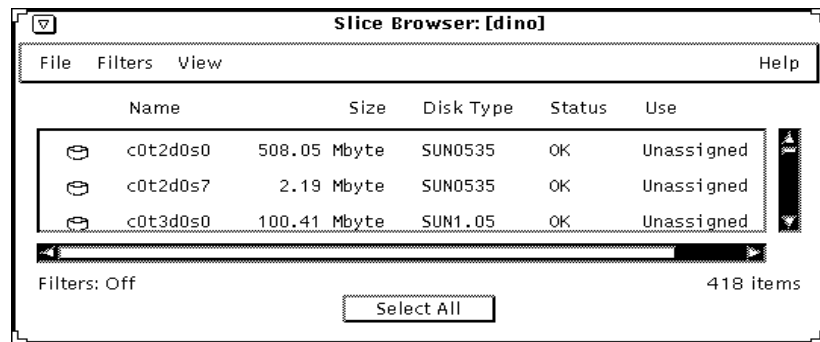
Click on the Concat/Stripe template. An unassigned and uncommitted Concat/Stripe template is opened. When you point to the template, the message line displays the metadevice name, size, use, and status.



Click here to open a Concat/Stripe template.

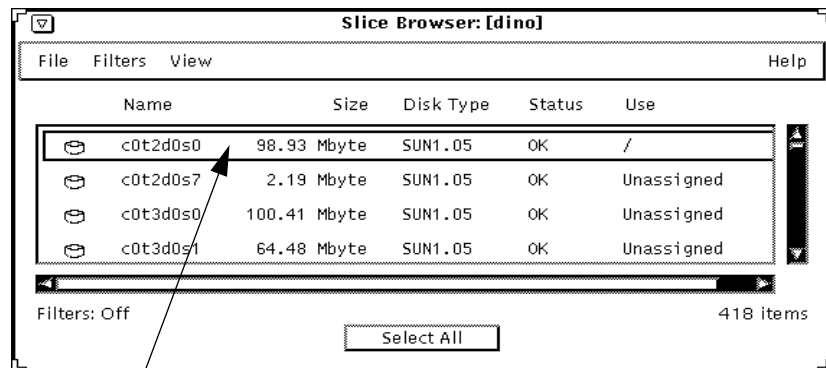
3. Open the Slice Browser.

Click on Slices on the button panel.



4. Locate the slice to be mirrored.

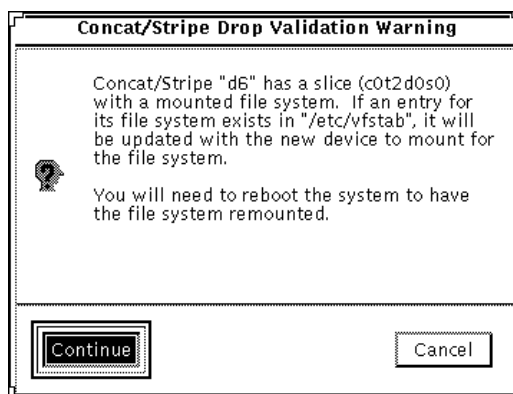
Slice c0t2d0s0, which contains the root file system, is selected in this example.



Click here to select the slice.

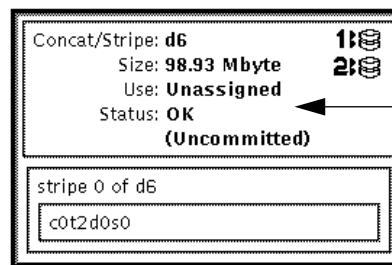
5. Drag c0t2d0s0 into the Concat/Stripe template.

Click on slice c0t2d0s0. Hold down the middle button. Then drag the slice to the Concat/Stripe d6 template. When the slice is dropped, a Warning dialog box is displayed. Click on the Continue button.



6. Commit the Concat/Stripe template.

Click inside the top rectangle of the Concat/Stripe object. Then click on Commit on the button panel.



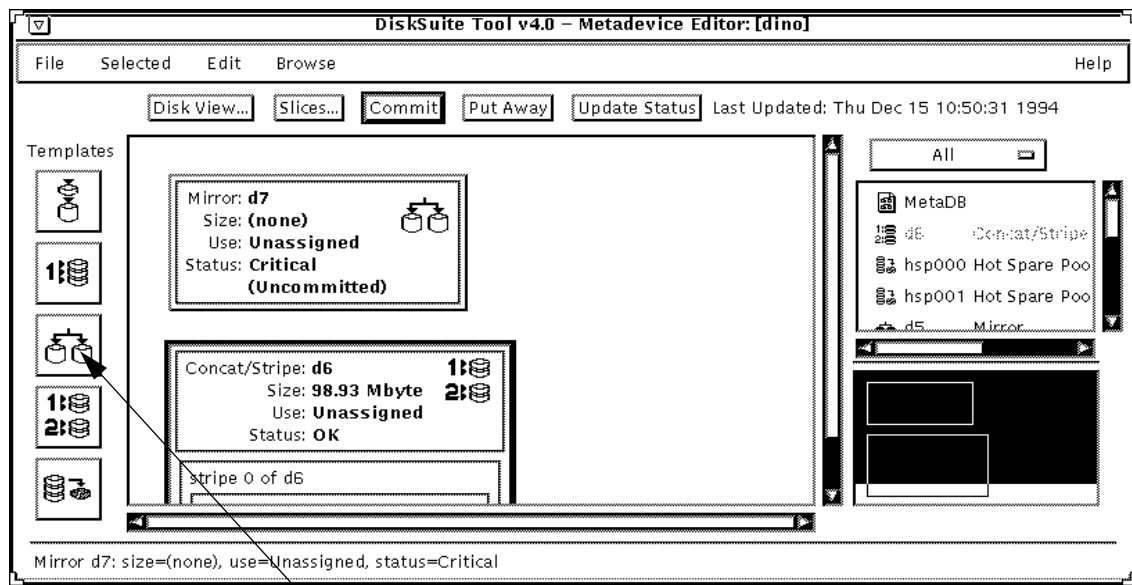
Click inside this rectangle.
Then click on the Commit button.

When the Commit action is invoked, the following warning dialog box is displayed. Click on the Really Commit button.



7. Open a Mirror template.

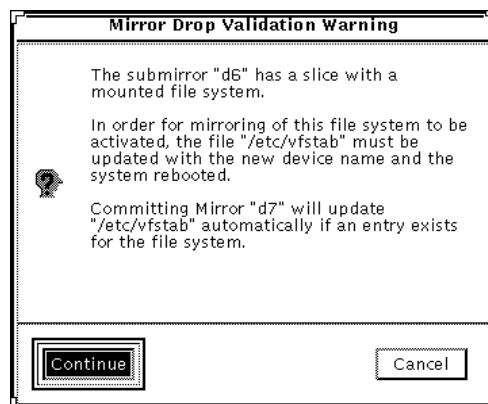
Click on the Mirror template. An unassigned and uncommitted Mirror template is opened.



Click here to open a Mirror template.

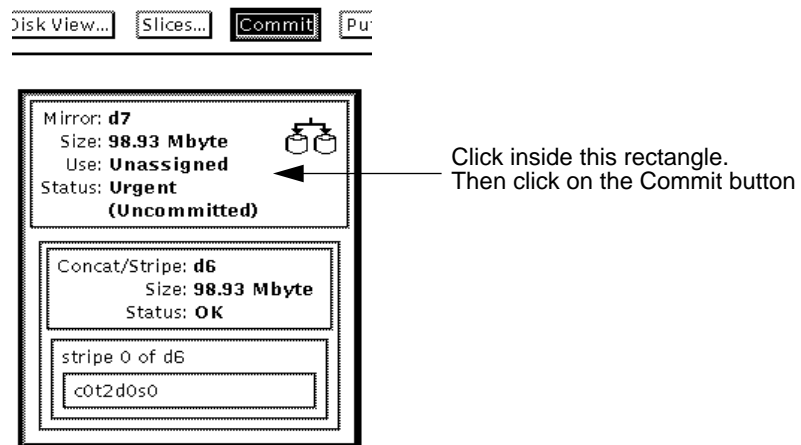
8. Drag the d6 Concat/Stripe into the Mirror template.

Click on the d6 Concat/Stripe object. Hold down the middle button. Drag the object into the Mirror template. The following warning dialog box is displayed. Click on the Continue button.



9. Commit the Mirror.

Click on the top rectangle in the Mirror template. Then click on Commit on the button panel.



When the Commit action is invoked, the following dialog box appears. Click on the Really Commit button.



10. Reboot the system.

You must reboot the system.

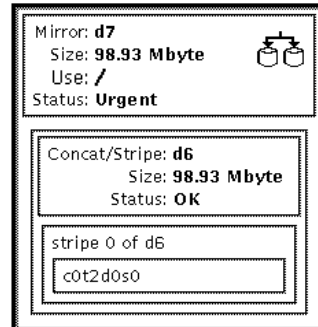
```
# reboot
```

11. After the system reboots, start DiskSuite Tool.

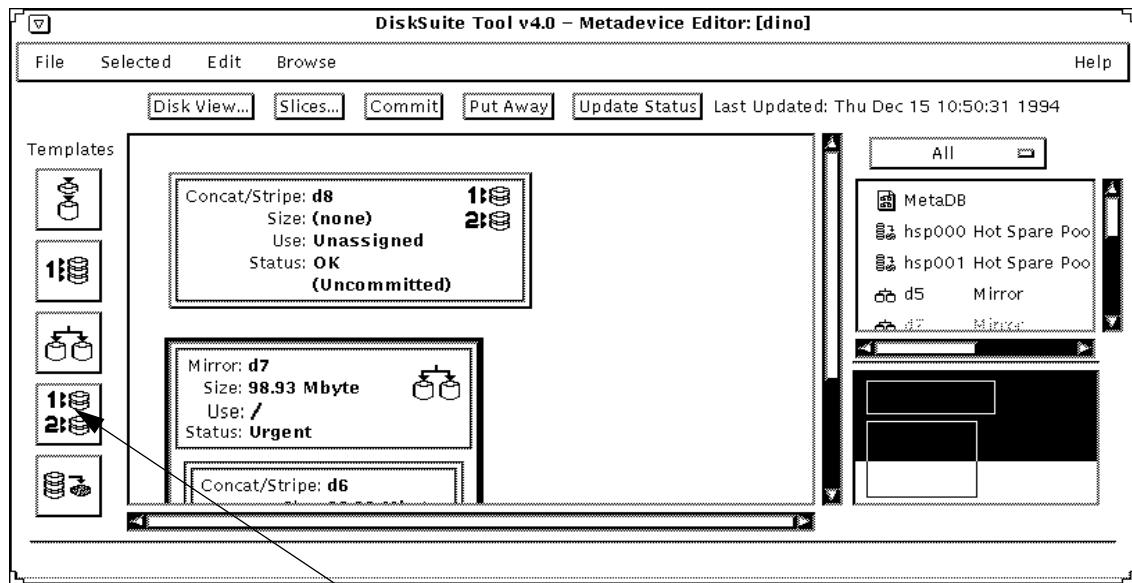
```
# metatool
```

12. Open the d7 Mirror object.

Scroll through the list of metadevices in the device list and double-click on the Mirror that contains the root file system (d7). The status remains Urgent because another submirror needs to be assigned.

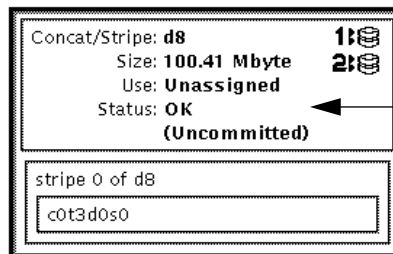


- 13. Open a Concat/Stripe template on the Metadevice Editor canvas.**
Click on the Concat/Stripe template.



Click here to open a Concat/Stripe template.

- 14. Populate the d8 Concat/Stripe with a slice.**
Select a slice in the Slice Browser that is the same size or larger than the root file system. The slice must be on a different disk and attached to a different controller. Point to the slice and press and hold down the middle button. Drag the slice into the d8 Concat/Stripe template.

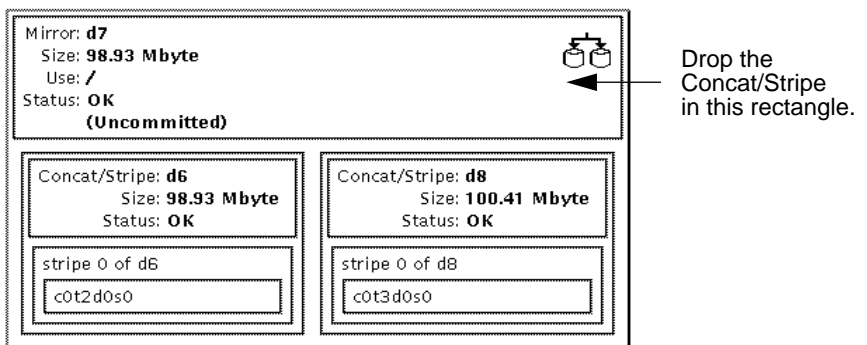


Drop the slice in this rectangle.

- 15. Commit the d8 Concat/Stripe.**
Click on the top rectangle of the Concat/Stripe template. Then click on the Commit button.

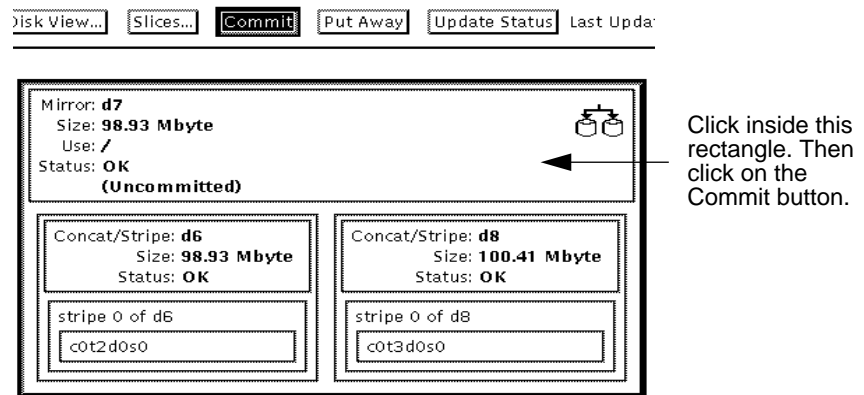
16. Drag the d8 Concat/Stripe into the d7 Mirror.

Point to the d8 Concat/Stripe object. Press and hold down the middle button. Drag the cursor into the Mirror template.



17. Commit the Mirror.

Click on the top rectangle of the Mirror object. Then click on Commit on the button panel.



If problems arise, refer to Chapter 5 of the *Solstice DiskSuite 4.0 Administration Guide*.

Unmirroring File Systems You Cannot Unmount

DiskSuite Tool does not support unmirroring the root (/) and swap file systems, or any other file systems (for example, /opt and /usr) that cannot be unmounted while the system is running.

Instead, use DiskSuite commands as documented in Chapter 5 of the *Solstice DiskSuite 4.0 Administration Guide*.

Recording and Booting From Alternate Root File Systems

If you mirrored root on either a SPARC or x86 system and the primary boot disk is lost, you must perform special steps to boot the system. A discussion of these procedures is given in Chapter 5 of the *Solstice DiskSuite 4.0 Administration Guide*.

Note – You must have recorded the path to the alternate root device when you are creating the mirror.

The method to use to obtain this path on a SPARC system is shown in the following example. This example assumes the root file system is being mirrored on slice c1t3d0s0.

```
# ls -l /dev/rdisk/c1t3d0s0
lrwxrwxrwx 1 root root 55 Mar 5 12:54 /dev/rdisk/c1t3d0s0 ->
../../../../devices/sbus@1,f8000000/esp@1,200000/sd@3,0:a
```

In the above SPARC example, you would record:

```
/sbus@1,f8000000/esp@1,200000/sd@3,0:a
```

Note – DiskSuite Tool users who are using system with open boot prom can use the OpenBoot `nvalias` command to define a “backup root” devalias for the secondary root mirror. For example:

```
ok nvalias backup_root /sbus@1,f8000000/esp@1,200000/sd@3,0:a
```

In the event of primary root disk failure, you then would only enter:

```
ok boot backup_root
```

The method to use to obtain this path on an x86 system is shown in the following example. This example assumes the root file system is being mirrored on slice `c1t0d0s0`.

```
# ls -l /dev/rdisk/c1t0d0s0
lrwxrwxrwx 1 root root  55 Mar 5 12:54 /dev/rdisk/c1t0d0s0 ->
    ../../devices/eisa/eha@1000,0/cmdk@1,0:a
```

In the above x86 example, you would record:

```
eisa/eha@1000,0/cmdk@1,0:a
```

Reconfiguring Submirrors

DiskSuite Tool provides several methods for reconfiguring submirrors with little or no disruption of service. The reconfiguring procedures are performed using the “Mirror Information Window” on page 137. The following subsections explain the steps used to attach or detach submirrors, bring submirrors online and offline, and to replace components within submirrors.

Replacement of a failed submirror can be performed in DiskSuite Tool by dropping a new metadvice on the failed component. The replacement can be for a single slice or the entire submirror (multiple slices).

To replace a failed slice, drag a new slice of equal or greater size onto the failed slice and release. Then commit the object. The instructions for replacing a slice in a mirror can be found in “Replacing and Enabling Submirror Components” on page 134.

Replacing Submirrors

To replace an entire submirror, you must first construct a new Concat/Stripe object that has a size equal to or greater than the size of the submirror that is being replaced. The new object is dropped onto the failed submirror. The mirror must be committed following the replacement.

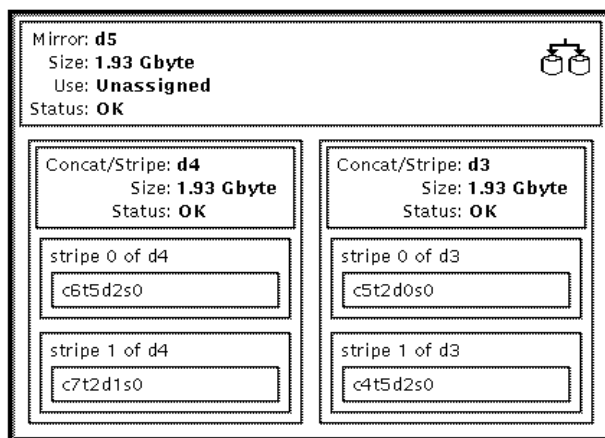
The procedure for replacement of an entire submirror includes the following steps:

1. Start DiskSuite Tool.

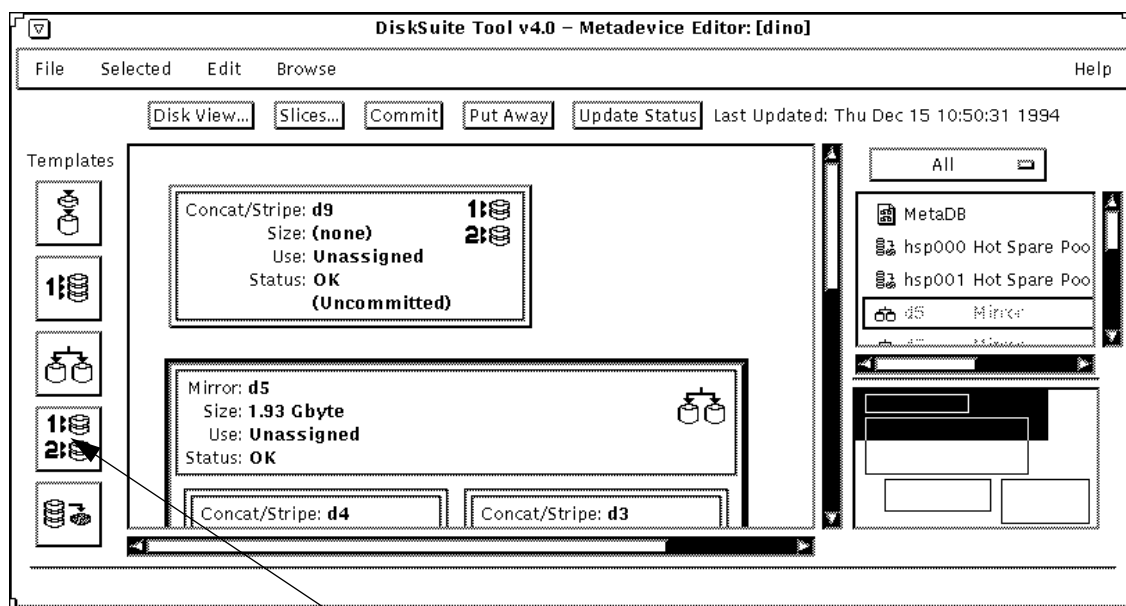
```
# metatool
```

2. Open the Mirror object on the Metadevice Editor canvas.

Point to the Mirror (in this example d5) in the device list and double-click.

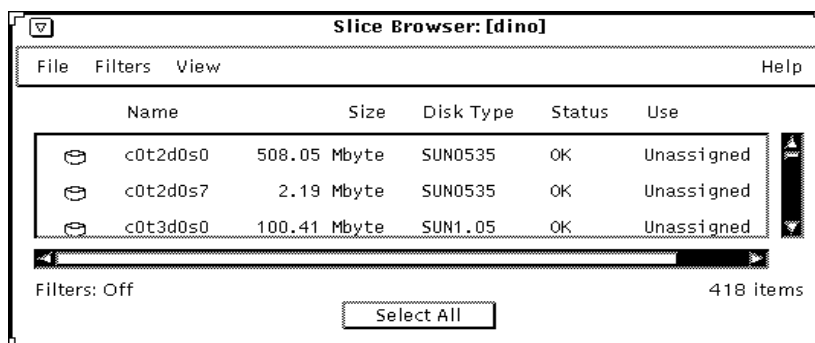


3. Open a Concat/Stripe template on the Metadevice Editor canvas.
Click on the Concat/Stripe template.



Click here to open a Concat/Stripe template.

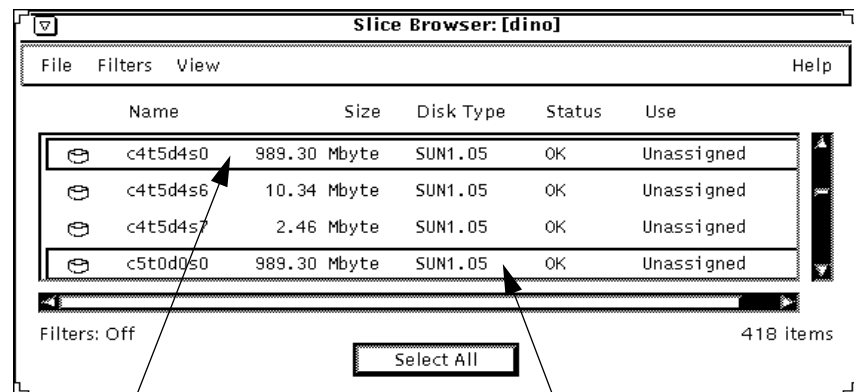
4. Open the Slice Browser.
Click on Slices on the button panel.



5. Select slices for the Concat/Stripe template.

Locate slices in the Slice Browser that when combined are equal to or greater than the size of the slices being replaced.

Note – The slices should be on different controllers than the slices in the other submirror. In this example, the Concat/Stripe objects in the d5 Mirror have each been populated with two 989.30-Mbyte slices to create two 1.93-Gbyte concatenations.

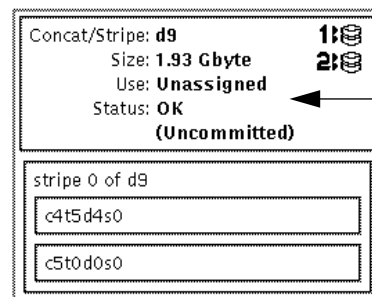


Click here to select the first slice.

Hold down the Control key and click here to select the second slice.

6. Drag the slices into the Concat/Stripe template.

Point to one of the two selected slices and press and hold down the middle button. Drag the slices to the Concat/Stripe d9 template.

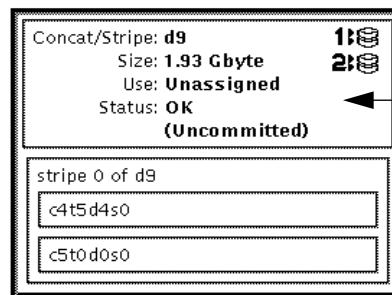


Drop the slices in this rectangle.

7. Commit the Concat/Stripe object.

Click inside the top rectangle of the Concat/Stripe. Then on the Commit button.

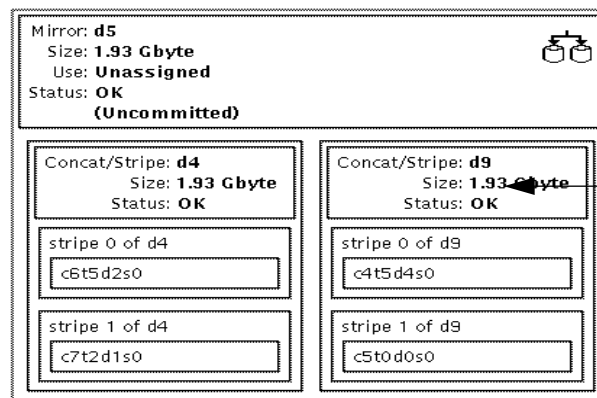
Disk View... Slices... **Commit** Put Away



Click inside this rectangle.
Then click on the Commit button.

8. Replace the submirror.

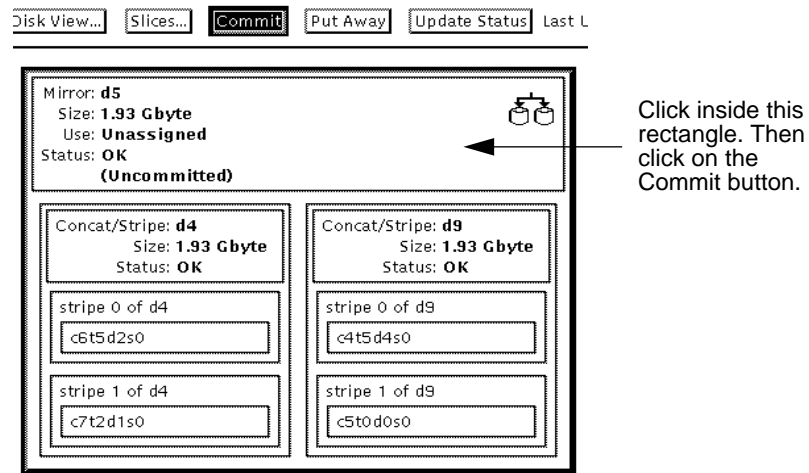
Point to the d9 Concat/Stripe object and press and hold down the middle button. Move the cursor to the top of the submirror that is being replaced. In this example, the d3 Concat/Stripe object is replaced with the d9 Concat/Stripe.



Drop replacement
submirror inside
this rectangle.

9. Commit the Mirror.

Click inside the top rectangle of the Mirror. Then click on the Commit button.



A resync will be initiated when you click on the Commit button.

Attaching and Detaching Submirrors

Concat/Stripe objects (metadevices) are attached to mirrors by dragging the object into the mirror template and committing the new mirror. Submirrors are detached by dragging the Concat/Stripe object from the mirror and then committing the mirror.

DiskSuite Tool automatically starts a resync operation on the new Concat/Stripe object after it has been attached to an existing mirror and the mirror is committed. The addition or replacement of a submirror occurs without an interruption of service to the component.

When Concat/Stripe objects are detached from the mirror and committed, they are no longer part of the mirror. Reads from and writes to the mirror no longer go to the detached submirror. The detached metadevice can be used for other purposes. To help protect your data, DiskSuite Tool does not allow the last remaining submirror of a mirror that is in use to be detached.

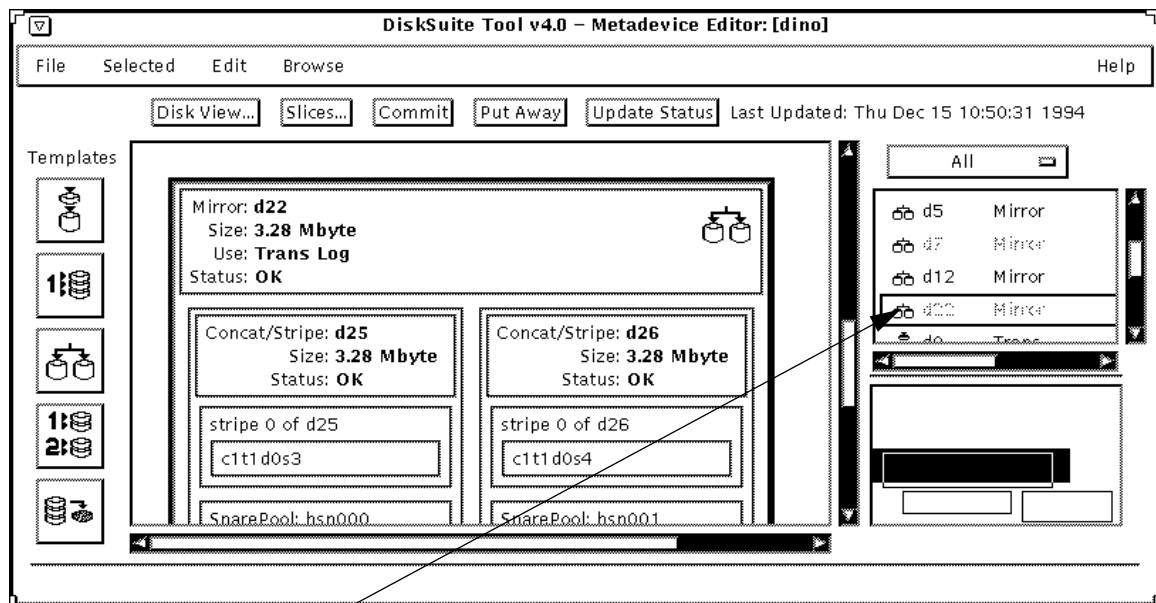
For example, a Concat/Stripe object could be detached (removed) when errors are being reported. To perform this procedure, use the following steps:

1. Start DiskSuite Tool.

```
# metatool
```

2. Open the Mirror on the Metadevice Editor canvas.

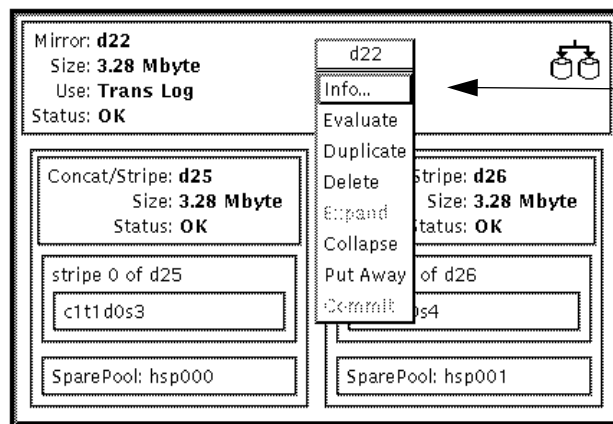
Locate the desired Mirror in the device list of the Metadevice Editor window. Point to the Mirror and double-click.



Double-click here to open the Mirror.

3. Bring up the Mirror Information window.

Point to the top of the Mirror object and press the right button. Select the Info option.



Point inside this rectangle and press the right button. Select the Info option.

4. Detach the submirror.

When the Mirror Information window opens, select the submirror that will be detached by pointing to it and clicking the left button. Click on the Remove button. (Refer to “Mirror Information Window” on page 137 for additional information about this window.)

Mirror Information: [dino]

Device Name: d22

Status: **OK**

Size: **3.28 Mbyte**

Use: **Trans Log**

Pass:

Read Option:

Write Option:

☒ Show Submirrors (2)

Device	Type	Size	Status
d25	Concat/Stripe	3.28 Mbyte	OK
d26	Concat/Stripe	3.28 Mbyte	OK

Device:

Click here to select the submirror.

Click here to remove.

Note – To attach a new submirror, you would open the Mirror Information window for the desired Mirror, enter the name of the device in the Device field, and click on the Attach button.

5. Commit the Mirror.

Click inside the top rectangle of the Mirror. Then click on the Commit button.

Placing Submirrors Online and Offline

DiskSuite Tool enables you to take submirrors offline and bring them back online. This functionality is useful when, for instance, one component in a physical SCSI chain fails. In this case, all other components on the chain could be taken offline while the broken component is replaced. After replacing the component, the other components in the SCSI chain can be brought back online.

When a submirror is offline, the mirror does not read from or write to that submirror. While the submirror is offline, DiskSuite Tool keeps track of all writes to the mirror. When the submirror is brought back online, the mirror and submirror are resynced. Writes are also directed to the submirror during the resync. Reads, however, come from a different submirror during the resync operation. Once the resync is complete, reads and writes are once again performed on the submirror previously taken offline.

Use the following steps to take a submirror offline.

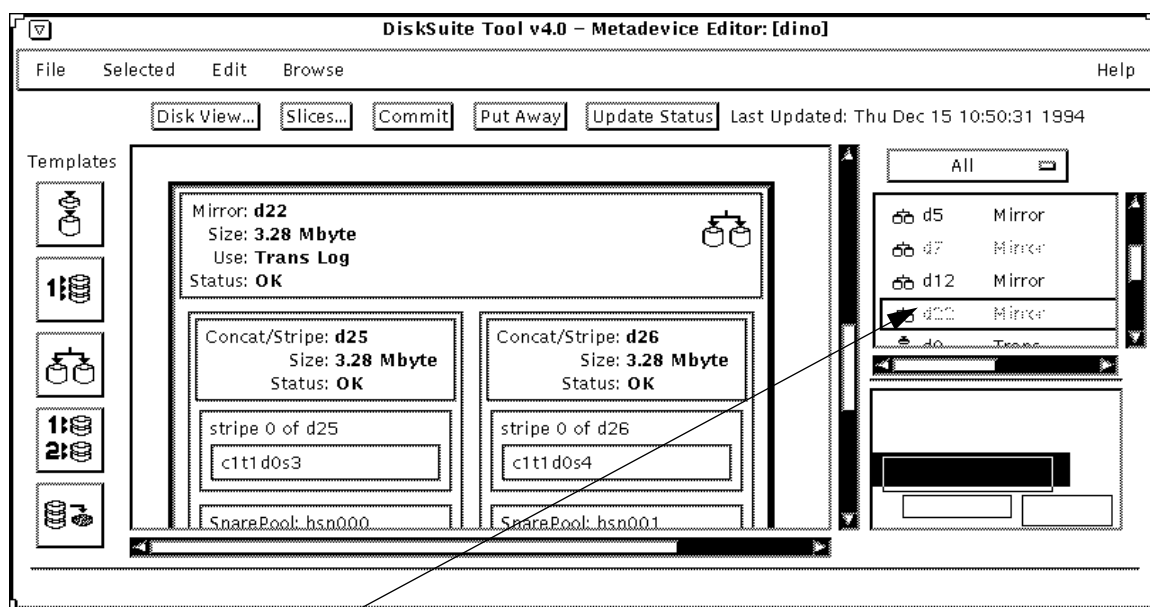
Note – You would follow the same procedure to bring a submirror back online. The difference is selecting the “Online” button on the Mirror Information window in Step 4.

1. Start DiskSuite Tool.

```
# metatool
```

2. Open the Mirror on the Metadevice Editor canvas.

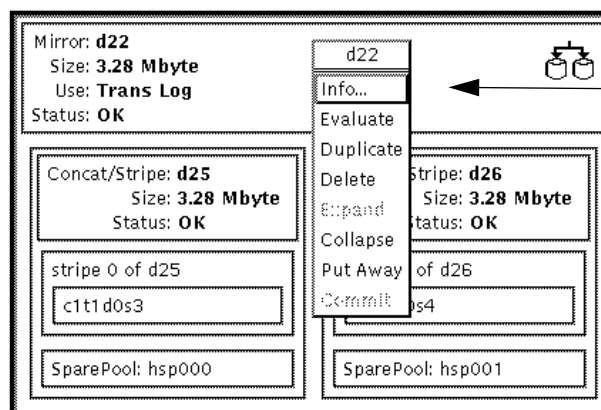
Locate the desired Mirror in the device list of the Metadevice Editor window. Point to the Mirror and double-click.



Double-click here to open the Mirror.

3. Bring up the Mirror Information window.

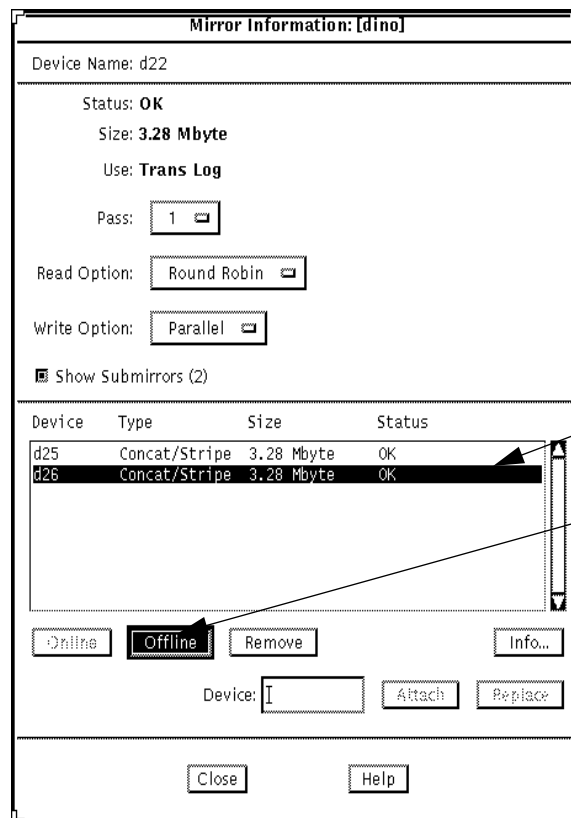
Point inside the top rectangle of the Mirror object and press the right button. Select the Info option.



Point inside this rectangle and press the right button. Select the Info option.

4. Take the submirror offline.

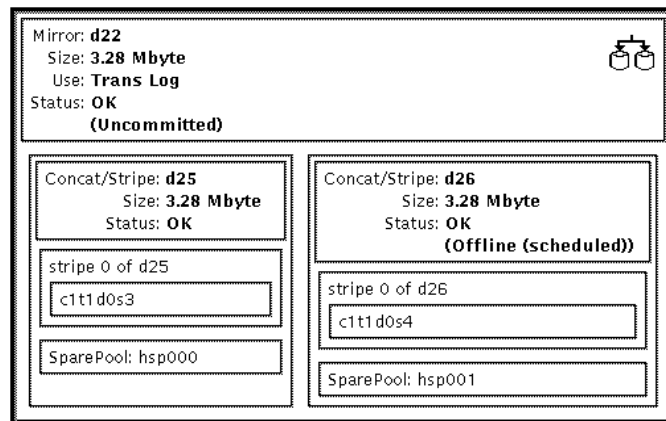
When the Mirror Information window opens, click on the submirror that will be taken offline. Then click on the Offline button. (Refer to “Mirror Information Window” on page 137 for additional information about using this window.)



Click here to select the submirror.

Click here to take the submirror offline.

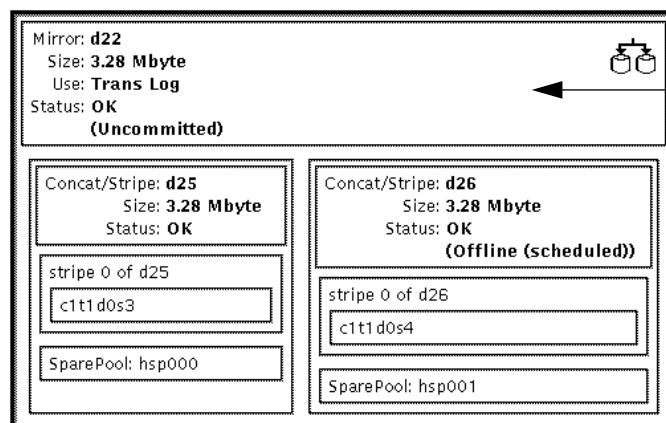
The status of the Concat/Stripe object becomes Offline (scheduled).



5. Commit the Mirror.

Click inside the top rectangle of the Mirror. Then click on the Commit button.

Disk View... Slices... **Commit** Put Away Update Status Last Updated: Thu



Click inside this rectangle. Then click on the Commit button.

Replacing and Enabling Submirror Components

DiskSuite Tool can be used to replace failed components within a submirror with a new component. When the component is replaced, a resync automatically starts, bringing the new component in sync with the rest of the mirror.

Note – Before using DiskSuite Tool to replace or enable submirror components, make sure the component is partitioned correctly. Refer to Chapter 5 of the *Solstice DiskSuite 4.0 Administration Guide* for additional information.

A component being used as a replacement in a submirror must be at least as large as the component it is replacing.

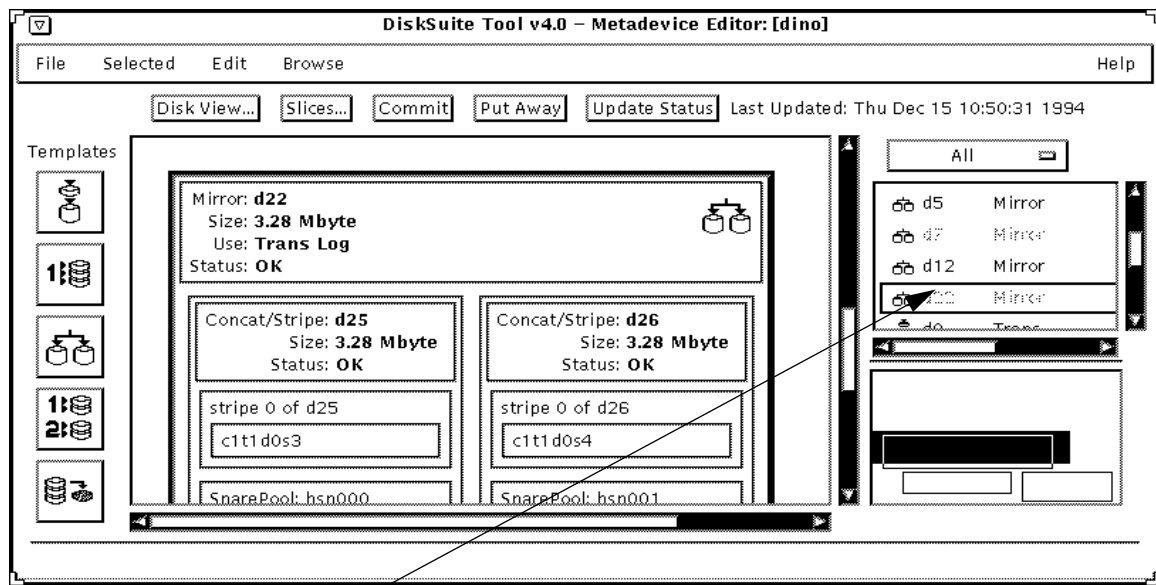
Use the following instructions to replace a failed or errored component in a mirror:

1. Start DiskSuite Tool.

```
# metatool
```


2. Open the Mirror on the Metadevice Editor canvas.

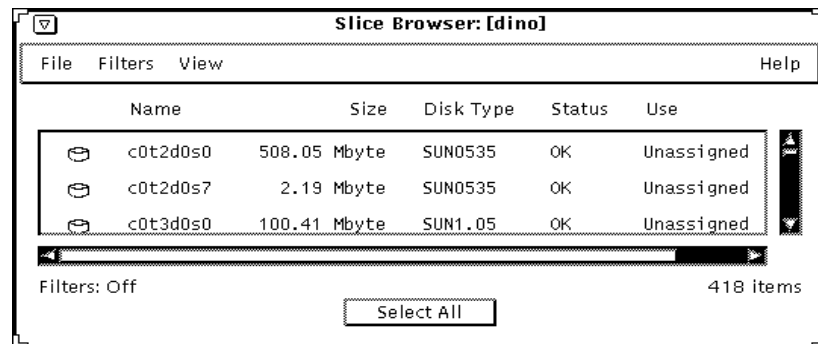
Locate the desired Mirror in the device list of the Metadevice Editor window. Point to the Mirror and double-click.



Double-click here to open the Mirror.

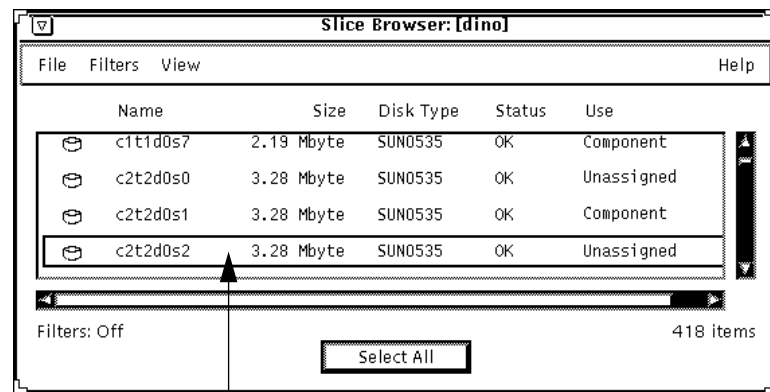
3. Open the Slice Browser.

Click on Slices on the button panel.



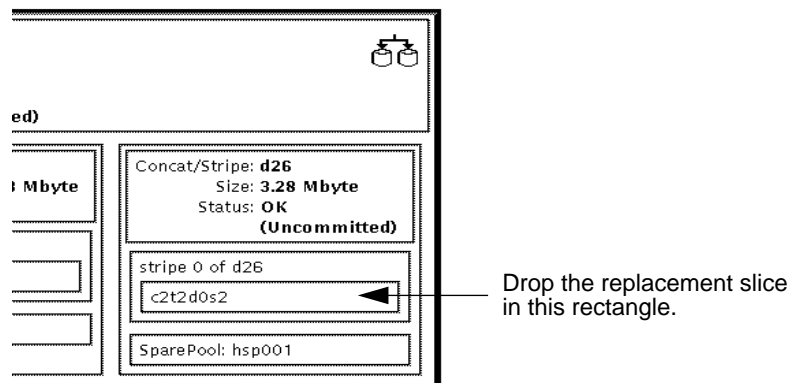
4. Select a slice to replace the failed or errored component.

Scroll through the list of devices in the Slice Browser and locate a slice that is the same size or larger than the failed or errored component.



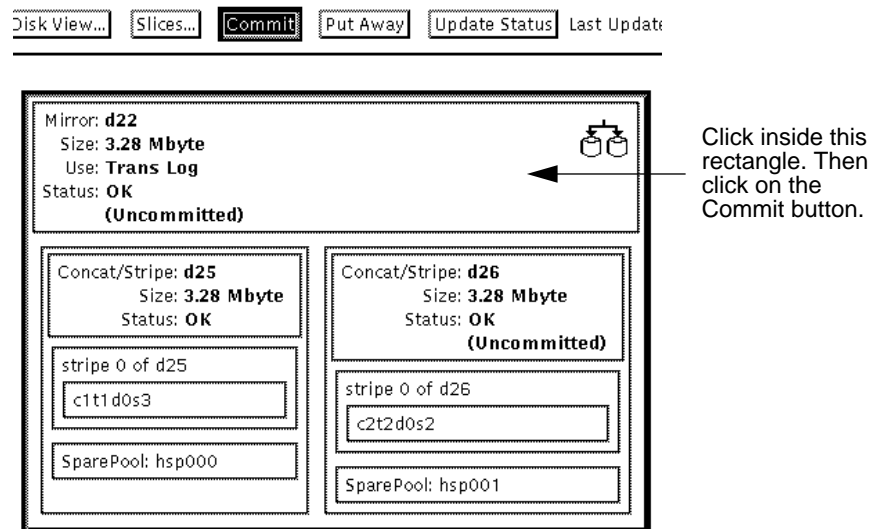
5. Replace the failed or errored component with the new slice.

Point to the new slice in the Slice Browser. Press down the middle button while moving the cursor to the rectangle that contains the slice you are replacing.



6. Commit the Mirror.

Click inside the top rectangle of the Mirror. Then click on the Commit button.



Using Mirrors for Online Backup

Mirroring enables you to perform online backups. Because each submirror is an exact copy of the file system, it can be taken offline and backed up to tape. This eliminates taking the system down to make a backup of the system.

You cannot perform online backups using DiskSuite Tool. Instead, use the steps in Chapter 5 of the *Solstice DiskSuite 4.0 Administration Guide*.

Mirror Information Window

DiskSuite Tool provides several options to optimize mirror performance. These options deal with the read and write policy for mirrors, and the order in which mirrors are resynched during reboot. You set these and other options using choices on the Mirror Information window, as shown in Figure 5-1. There are three ways to display the Mirror Information window:

- Point to the mirror in the Metadevice Browser and double-click. The mirror will open on the Metadevice Editor's canvas and the Mirror Information window is displayed.
- If the mirror object is on the Metadevice Editor's canvas, point inside the mirror and bring up the mirror menu. Select the Info choice.
- Double-click on a mirror object on the Metadevice Editor's canvas.

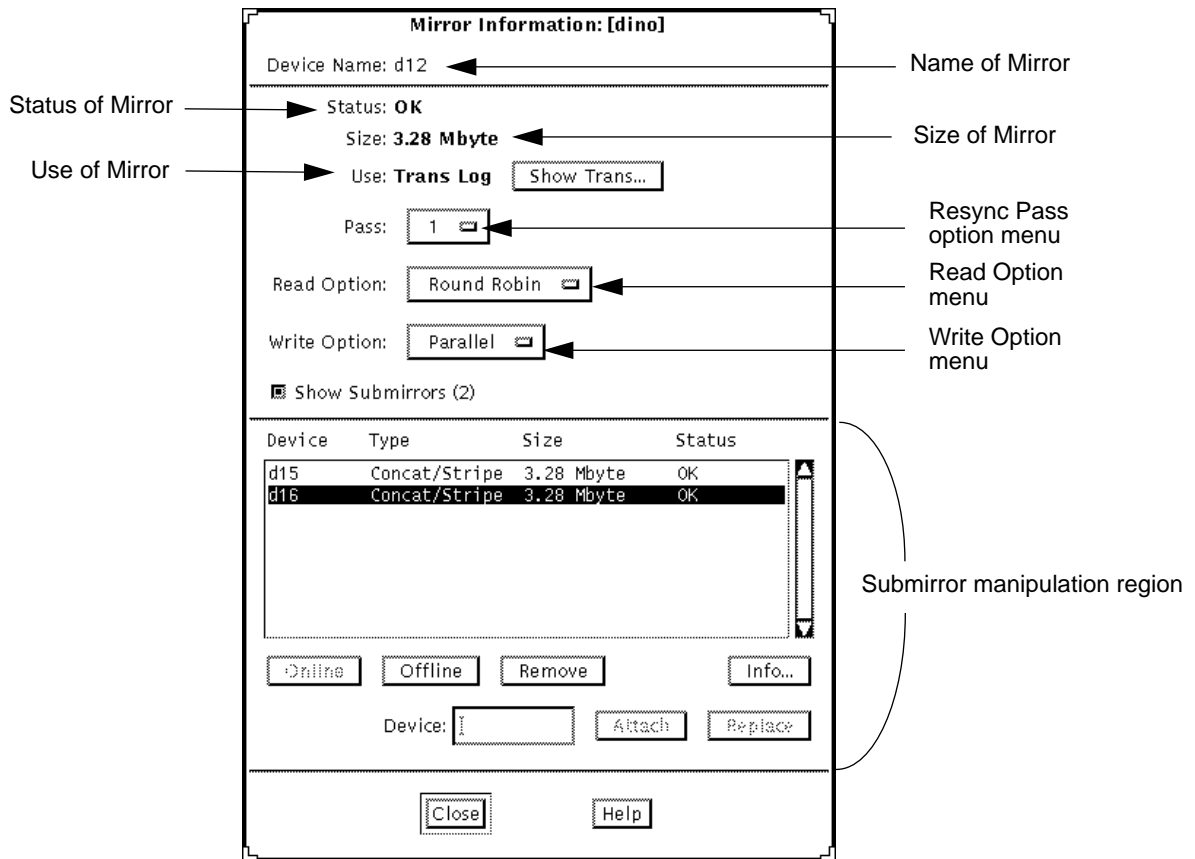


Figure 5-1 Mirror Information Window

The Mirror object must be committed before the policy changes take effect.

Table 5-1 lists the functionality associated with the regions of the Mirror Information window.

Table 5-1 Mirror Information Window Functionality

Field	Functions
Device name	The metadvice name of the mirror.
Status	Description of the mirror's status. See "Checking the Status of Mirrors" on page 141 for a description of the values that can appear here.
Size	The size of the mirror.
Use	Shows how the mirror is currently used, for example, file system, swap, or shared log. If the use is shared log, a button labeled Show Trans is displayed. The Show Trans button opens a Sharing Information window that shows the Trans devices that share the Mirror.
Show slices	This toggle button enables you to turn on and off the slice manipulation region. The number of slices in the stripe are shown in parentheses on the button.
Pass	A pass number in the range 0-9 can be assigned to a mirror using the Pass button menu. The pass (resync) number determines the order in which that mirror is resynced during a system reboot. The default is 1. Smaller pass numbers are resynced first. If 0 is chosen, the resync is skipped. A 0 should only be used for mirrors mounted as read-only. If different mirrors have the same pass number, they are resynced concurrently.

Table 5-1 Mirror Information Window Functionality

Field	Functions
Read Option	<p>There are three kinds of read options associated with mirrors: Round Robin, Geometric, and First. The default read option is Round Robin, also called balanced load.</p> <p>When set to Round Robin, all reads are made in a round robin order from all the submirrors in the mirrors. That is, the first read comes from the first submirror, the next read comes from the second submirror, and so forth.</p> <p>The Geometric option provides faster performance on sequential reads or when you are using disks with track buffering. Geometric reads allow read operations to be divided among submirrors on the basis of a logical disk block address. For instance, with a three-way submirror the disk space on the mirror is divided into three (equally sized) logical address ranges. Reads from the three regions are then performed by separate submirrors (for example, reads to the first region are performed by the first submirror).</p> <p>The First option specifies reading from only the first submirror. This would be specified only if you have a second submirror that has poor read I/O characteristics.</p>
Write Option	<p>A button that enables you to set parallel or serial writes to the submirror. Parallel writes are the default action of the metadisk driver, meaning the writes are dispatched to all submirrors simultaneously.</p>

Table 5-1 Mirror Information Window Functionality

Field	Functions
Submirror manipulation region	<p>The following functionality is available in this region:</p> <p>Show Submirrors – This toggle button enables showing or hiding the list of submirrors.</p> <p>Scrolling List – Shows submirrors included in the mirror. The information in this region includes the name, type, size, and status. Click on the submirror to select it. When submirrors are selected, actions can be performed on them.</p> <p>Online – Enables you to bring selected submirrors back online. This button is active only when the selected submirror is offline.</p> <p>Offline – Enables you to take selected submirrors offline. This button is active only when the selected submirror is online.</p> <p>Remove – Enables you to detach the selected submirrors.</p> <p>Info – Opens the Concat Information window for the selected submirror.</p> <p>Device – Enables you to specify a new submirror in the field to attach or replace. The field is cleared when you click on the Attach or Replace buttons.</p> <p>Attach – Adds the specified submirror. This button is active only when a submirror or device is entered in the Device field.</p> <p>Replace – Replaces the selected submirror with the submirror entered in the field. This button is active only when a submirror or device is entered in the field and one in the list is selected.</p>

Checking the Status of Mirrors

There are three ways to check the status of submirrors using DiskSuite Tool. The methods are:

- Move the mirror from the device list and check the Status field on the submirrors.
- Bring up the Mirror Information window and check the Status field for each of the submirrors.
- Bring up the Problem List window, which is available using the Browse menu on the menu bar of the Metadevice Editor window. If a problem exists, a detailed description of the problem is given.

The Status fields of Mirror objects use the status keywords explained in Table 5-2.

Table 5-2 Mirror Status Keywords

Keyword	Meaning
OK	The submirror has no errors and is functioning correctly.
Component Error	The submirror has a slice that has errors.
Resyncing	The submirror is actively being resynced. An error has occurred and been corrected, the submirror has just been brought back online, or a new submirror has been added.
Attaching	The submirror is being attached.
Attached (resyncing)	The submirror is being resynced after the attach occurred.
Offlined	The submirror is offline.
Errored	The submirror has an error.
Online (scheduled)	The submirror will be brought online the next time you click on the Commit button.
Attention	There is a problem with the submirror, but there is no immediate danger of losing data. There are still two copies of the data, but a problem exists somewhere in the configuration.
Urgent	A submirror has failed and only one copy of the data exists. You are only one failure away from losing data.
Critical	Data has potentially been corrupted.

UFS Logging



This chapter provides information on how to use Solstice DiskSuite Tool (`metatool(1M)`) to create and manipulate UFS logging (trans) devices. Refer to Chapter 3, “Overview of DiskSuite Tool,” for information on using DiskSuite Tool.

Use the following table to locate specific information.

<i>Overview of UFS Logging</i>	<i>page 143</i>
<i>How to Set Up UFS Logging on an Existing File System</i>	<i>page 146</i>
<i>How to Share a Log Between File Systems</i>	<i>page 150</i>
<i>Removing UFS Logging</i>	<i>page 154</i>
<i>Logging a File System That You Cannot Unmount</i>	<i>page 157</i>
<i>Removing UFS Logging From a File Systems You Cannot Unmount</i>	<i>page 162</i>
<i>Using Metadevices and Mirrors as Trans Components</i>	<i>page 162</i>
<i>Trans Information Window</i>	<i>page 163</i>
<i>Checking the Status of Trans Metadevices</i>	<i>page 165</i>
<i>Recovering From Device Errors and File System Panics</i>	<i>page 165</i>

Overview of UFS Logging

DiskSuite’s UFS logging facility speeds up reboots, provides faster local directory operations, and decreases synchronous disk writes. UFS file system updates are safely recorded in a log before they are applied to the file system itself.

UFS file systems are checked at boot time because unscheduled system downtime can interrupt file system updates. These partially completed updates can leave inconsistencies in a file system. Mounting a file system without first checking it and repairing any inconsistencies can cause panics or data corruption. Checking consistency for large file systems can be a time-consuming process. With the UFS logging feature, file systems do not have to be checked at boot time because the changes from unfinished system calls are discarded.

A pseudo device, called the trans device, manages the contents of the log. Like other metadevices, the trans device behaves the same as an ordinary disk partition. The trans device is made up of two subdevices: the logging device and the master device. These can be disk partitions, or metadevices, but not trans devices.

The logging device can be shared by several file systems. The logging device contains the log. This log is a sequence of records, each of which describes a change to a file system. The master device contains the file system itself. The master device can either contain a file system initially, or you can create a file system on the trans device. Logging begins automatically when the trans device is mounted.

How UFS Logging Works

You set up and configure a trans device using DiskSuite Tool. The configuration of the device and other state information is stored in the metadevice state database. The dynamic concatenation facility provides dynamic concatenation of both the master and logging devices.

Trans devices have the same naming convention used for other metadevices.

After a trans device is configured, it can be used just as if it were a physical partition. This means it can be used as a device that is as large as one terabyte in size. A file system can be created on the trans device if the master device doesn't already have a file system. Most UNIX disk utilities will work normally on trans devices, with the exception of `format(1M)`.

In addition, all usual file system operations can be performed on a trans device. The following list offers examples of file system operations:

- Mount (`mount(1M)`) the trans device on a directory
- Unmount (`umount(1M)`) a mounted trans device

- Copy files to the mounted trans device
- Read and write files from and to the mounted trans device
- Run `ufsdump(1M)` and `ufsrestore(1M)` on the trans device

The following sections provide answers to some of the questions you'll need to resolve before you begin setting up UFS logging on your system.

Space Requirements for Logs

The minimum size for a logging device is one Mbyte of disk space. Larger logs allow for more simultaneous file system operations per second. As a general rule, you need about one Mbyte of log space for every 100 Mbytes of file system space being logged.

Note – Only the first 64 Mbytes allocated to a log are used. Larger logs will contain unused space.

Selecting Which File Systems Should Be Logged

It is possible to log any UFS file system with the exception of `root (/)`. The `root` file system cannot be a trans device. In addition, it is probably not necessary to log small file systems with mostly read activity.



Caution – Logging must be disabled for `/usr`, `/var`, `/opt`, or any other file system used during a Solaris upgrade or installation.

In general, you will want to log your largest UFS file systems and the UFS file systems whose data is changed most frequently.

All logged file systems can share the same log. For performance considerations, however, the file systems with the heaviest loads should have separate logs.

Note – You should mirror all logs. Losing the data in a log because of device errors can leave a file system in a corrupt state.

Determining the Location of Logs

Before you begin the procedures for setting up UFS logging, you should determine where to place the logs. You have two options for placement of logs:

- On unused partitions
- On the partitions containing the state databases

The second of these options can be a practical solution if you are unable to commit an unused partition to logging. Refer to Chapter 2 and Chapter 10 in the *Solstice DiskSuite 4.0 Administration Guide* for additional information.

How to Set Up UFS Logging on an Existing File System

After you have determined which file systems you want to log and where the log will be located, follow the steps in this section to set up UFS logging.

Note – The procedure for setting up UFS logging on slices that are not occupied by a UFS is the same as the following procedure. It is not necessary to unmount the file system. However, you must initialize the file system (using `newfs(1M)`) before mounting the file system.

The following instructions tell you how to set up the master for UFS logging on an existing file system. In this example, the file system is named `/abcf`:

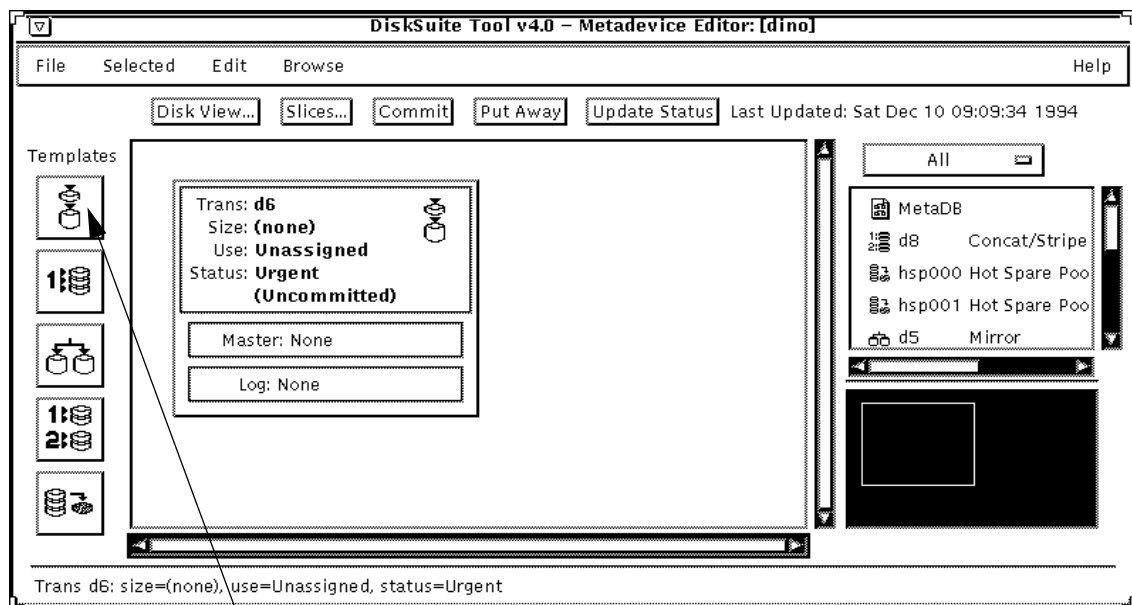
- 1. As root, use the `umount(1M)` command to unmount `/abcf`.**

```
# umount /abcf
```

- 2. Start DiskSuite Tool.**

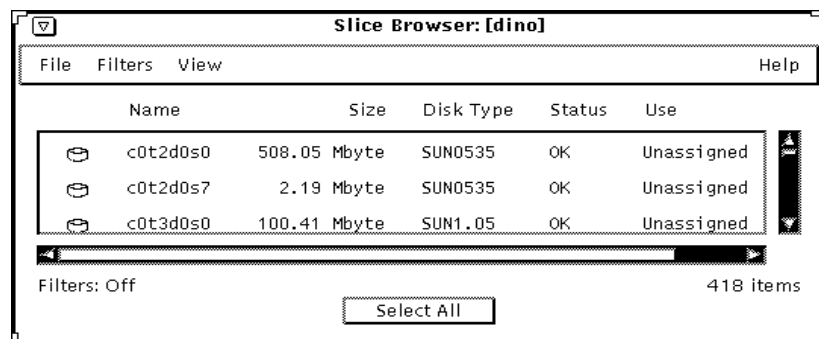
```
# metatool
```

3. **Open a trans device template on the Metadevice Editor canvas.**
Click on the Trans template. An unassigned and uncommitted Trans template is opened. When you point to the template, the message line displays the Trans device name, size, use, and status.



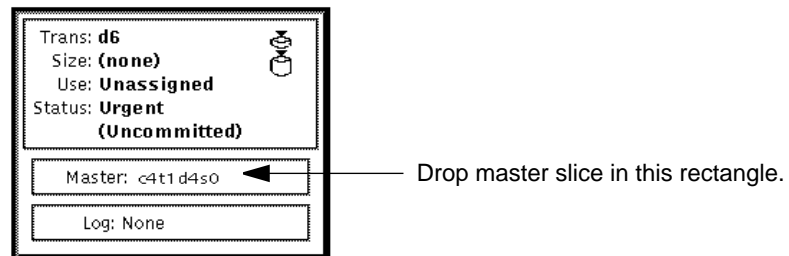
Click here to open a Trans template.

4. **Open the Slice Browser.**
Click on Slices on the button panel.



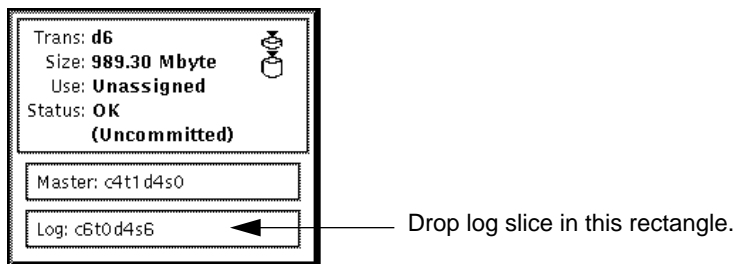
5. Select the slice that will contain the UFS logging master.

Click on the slice that will contain the UFS logging master. Then point to the slice, press the middle button, and drag the slice to the master rectangle in the Trans template. In this example, c4t1d4s0 is selected.

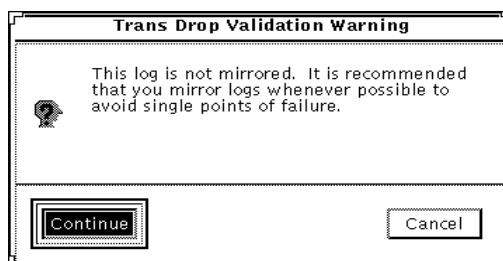


6. Select the slice that will contain the UFS log.

Click on the slice that will contain the UFS log. Point to the slice, press the middle button, and drag the slice to the log rectangle of the Trans template. In this example, slice c6t0d4s6 is selected.

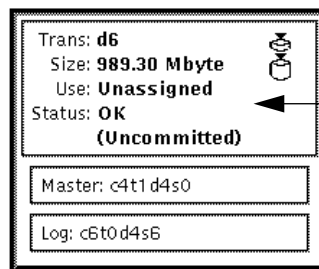
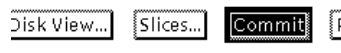


If the log slice is not mirrored, the following warning dialog box is displayed.



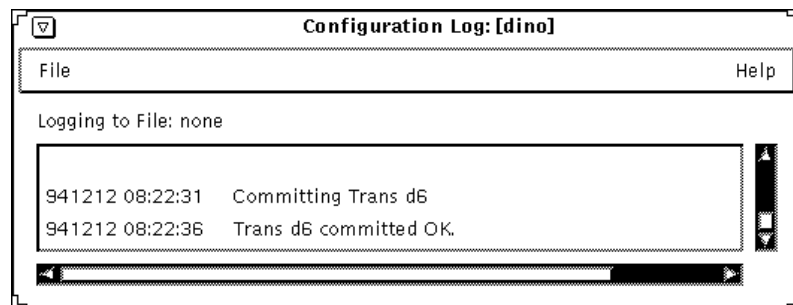
7. Commit the Trans device.

Click on the top rectangle of the Trans template. Then click on the Commit button.



Click inside this rectangle.
Then click on the Commit button.

The Configuration Log displays the message that the Trans device has been committed.



8. Edit the `/etc/vfstab` file.

After you complete the previous steps, you must change the entry for `/abcf`s in the `/etc/vfstab` file. For example, the following line:

```
/dev/dsk/c4t1d4s0 /dev/rdisk/c4t1d4s0 /abcf s ufs 4 yes -
```

should be changed to read:

```
/dev/md/dsk/d6 /dev/md/rdisk/d6 /abcf s ufs 4 yes -
```

9. Mount the file system.

Invoke the `mount(1M)` command and mount the `/abcf`s file system.

```
# mount /abcf
```

`/abcf`s is now a logging file system. When you reboot the system, `fsck(1M)` will notice that this is a logging file system and will not check it. Instead, `fsck` will display:

```
/dev/md/rdisk/d6: is logging.
```

How to Share a Log Between File Systems

DiskSuite Tool supports sharing logs between several or all file systems. This section contains the instructions for sharing a log between file systems.

In this example, the `/abcf`s file system that was logged in the previous subsection and the `/xyz`f file system will share the same log. The two file systems will have different masters.

The following instructions tell you how to set up `/xyz`f to share the same log with `/abcf`s:

1. Unmount the file system.

Invoke the `umount` command and mount the `/xyz`f file system.

```
# umount /xyz
```

2. Start DiskSuite Tool.

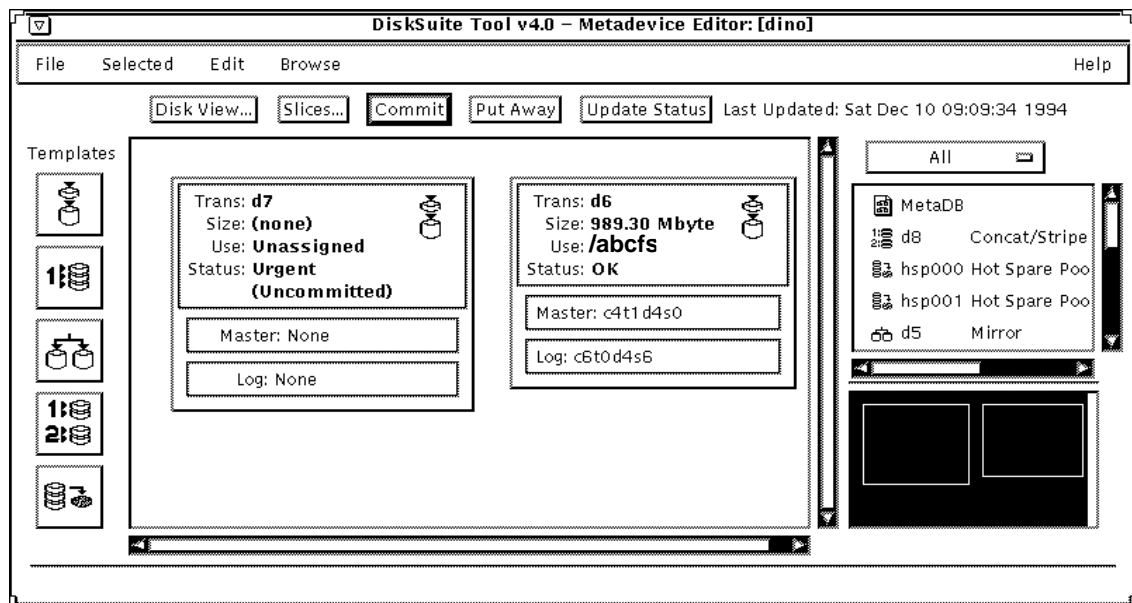
```
# metatool
```

3. Open a Trans device template on the Metadevice Editor canvas.

Click on the Trans template. In this example, `d7` is the Trans device that opens.

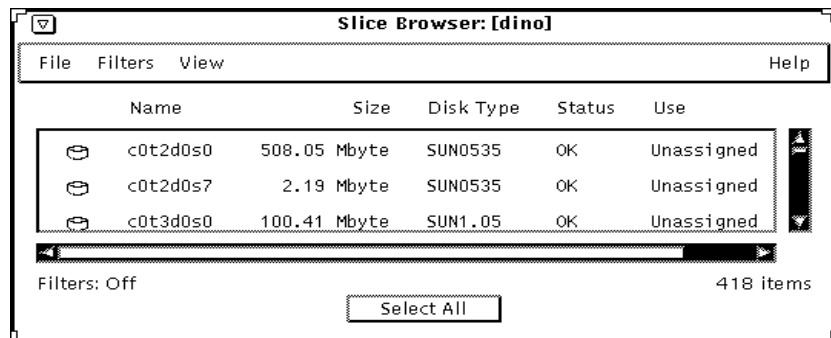
4. Open the d6 Trans device.

Point to the d6 Trans device in the device list of the Metadevice Editor window and double-click the left button.



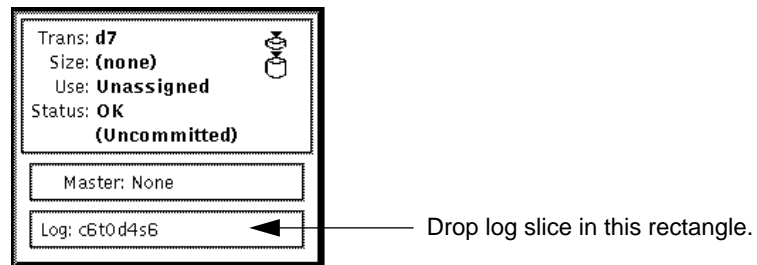
5. Open the Slice Browser.

Click on Slices on the button panel.

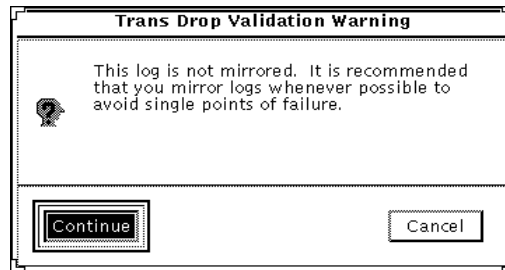


6. Select the slice that will contain the UFS log.

Click on the slice that will contain the UFS log. Then point to the slice, press the middle button, and drag the slice to the log rectangle in the d7 Trans template. In this example the same log used by the d6 Trans device is selected (c6t0d4s6).

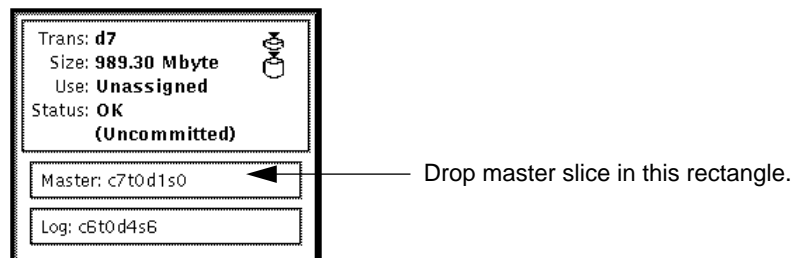


If the log slice is not mirrored, the following warning dialog box is displayed.



7. Select the slice that will contain the d7 UFS master.


Click on the slice that will contain the UFS master. Point to the slice, press the middle button, and drag the slice to the master rectangle in the Trans template. In this example, c7t0d1s0 is selected.



8. Commit the d7 trans device.


Click on the Trans template. Then click on the Commit button.


Disk View... Slices... **Commit** P

Trans: d7 Size: 989.30 Mbyte Use: Unassigned Status: OK (Uncommitted)	
Master: c7t0d1s0	
Log: c6t0d4s6	

Click inside this rectangle.
Then click on the Commit button.

The Trans devices now share the same log, c6t0d4s6.

Trans: d7 Size: 989.30 Mbyte Use: Unassigned Status: OK	
Master: c7t0d1s0	
Log: c6t0d4s6	

Trans: d6 Size: 989.30 Mbyte Use: /abcf Status: OK	
Master: c4t1d4s0	
Log: c6t0d4s6	

Shared log device c6t0d4s6

9. Edit the /etc/vfstab file.

After you complete the previous steps, you must change the entry for /xyzfs in the /etc/vfstab file. For example, the following line:

```
/dev/dsk/c7t0d1s0 /dev/rdisk/c7t0d1s0 /xyzfs ufs 4 yes -
```

should be changed to read:

```
/dev/md/dsk/d7 /dev/md/rdisk/d7 /xyzfs ufs 4 yes -
```

10. Mount the file system

Invoke the `mount` command and mount the `/xyzfs` file system.

```
# mount /xyzfs
```

`/xyzfs` is now a logging file system. When you reboot the system, `fsck` will notice that `/abcf`s and `/xyzfs` are logging file systems and will not check them. Instead, `fsck` will display:

```
/dev/md/rdisk/d6: is logging.  
/dev/md/rdisk/d7: is logging.
```

Removing UFS Logging

This subsection contains the instructions for removing a log from a file system by using DiskSuite Tool.

To remove UFS logging from the `/abcf`s file system, perform the following steps:

1. Unmount the `/abcf`s file system.

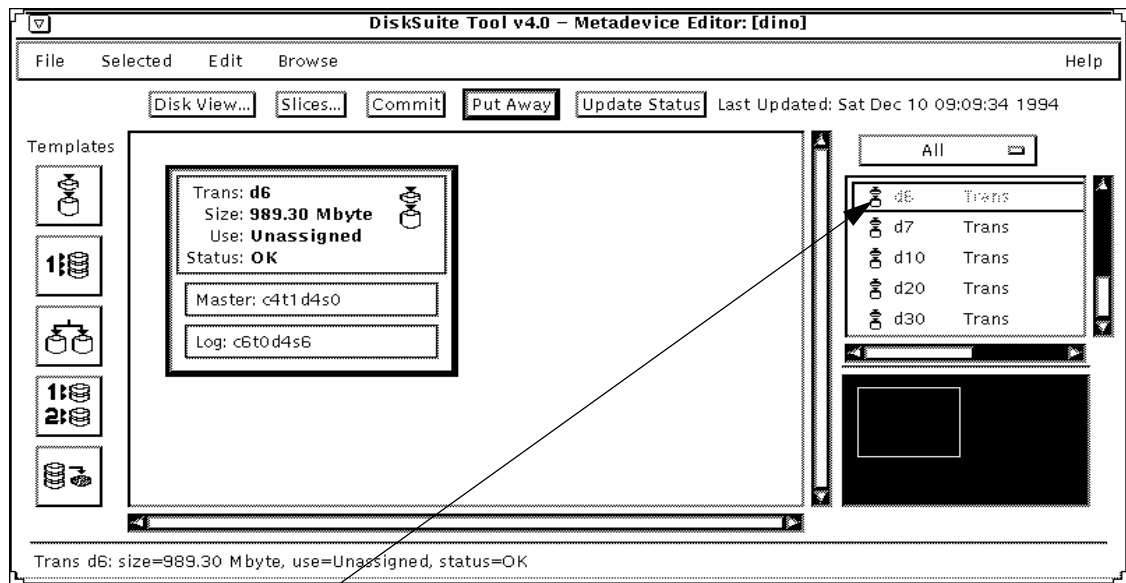
Invoke the `umount` command and unmount the `/abcf`s file system.

```
# umount /abcf
```

2. Start DiskSuite Tool.

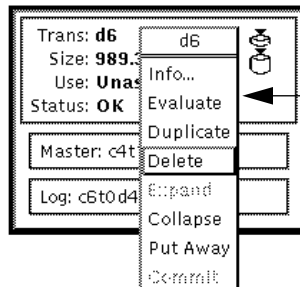
```
# metatool
```

3. Open the d6 Trans device on the Metadevice Editor window's canvas. Point to the Trans device in the device list and double-click.



Double-click here to open the Trans device.

4. Delete the d6 Trans device. Point inside the top rectangle of the d6 Trans device and press the right button. Select the Delete option.



Point inside this rectangle and press the right button. Select the Delete option.

5. Confirm the deletion of the d6 Trans device.

Click on the Really Delete button.



6. Edit the `/etc/vfstab` file to remove the d6 Trans device.

In this case, you would change the line:

```
/dev/md/dsk/d6      /dev/md/rdisk/d6      /abcfs  ufs  5  yes  -
```

to read:

```
/dev/dsk/c4t1d4s0  /dev/rdisk/c4t1d4s0  /abcfs  ufs  5  yes  -
```

7. Run the `fsck` command.

Because `/abcfs` is no longer a logging file system, you must run `fsck` before you can mount it. The following output is generated when you run `fsck`:

```
# fsck /dev/rdisk/c4t1d4s0
FILE SYSTEM STATE IN SUPERBLOCK IS WRONG; FIX? y
```

8. Mount the file system.

Invoke the `mount` command on the `/abcfs` file system.

```
# mount /abcfs
```

The `/abcfs` file system is no longer logging.

Logging a File System That You Cannot Unmount

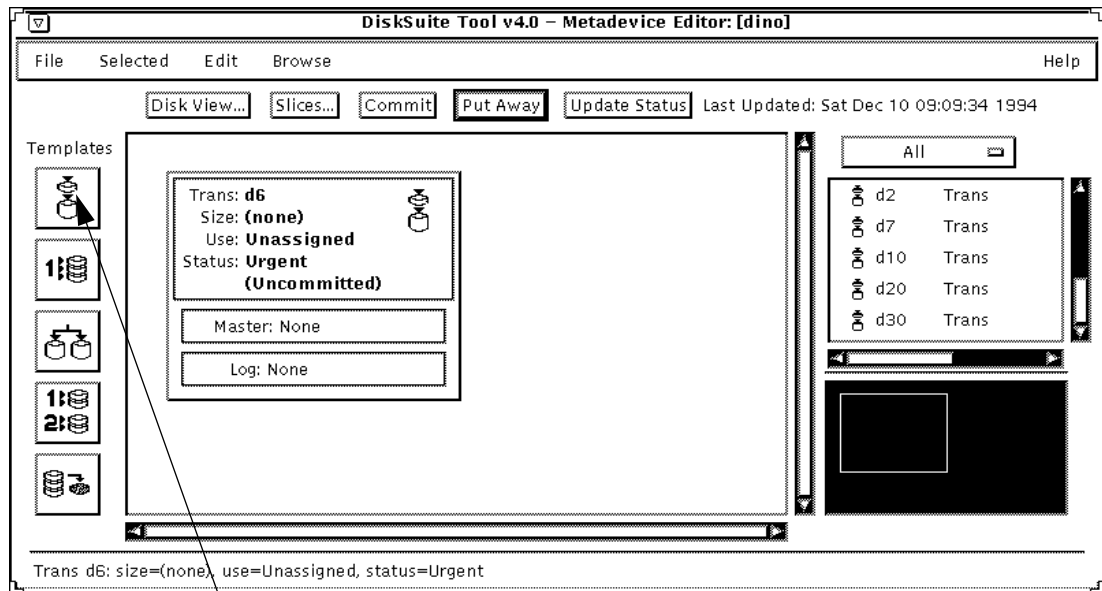
DiskSuite Tool supports UFS logging on file systems that cannot be unmounted while the system is running, such as `/var`, `/opt`, or `/usr`.

The following example provides step-by-step instructions for creating a UFS log on the `/var` file system.

1. Start DiskSuite Tool.

```
# metatool
```

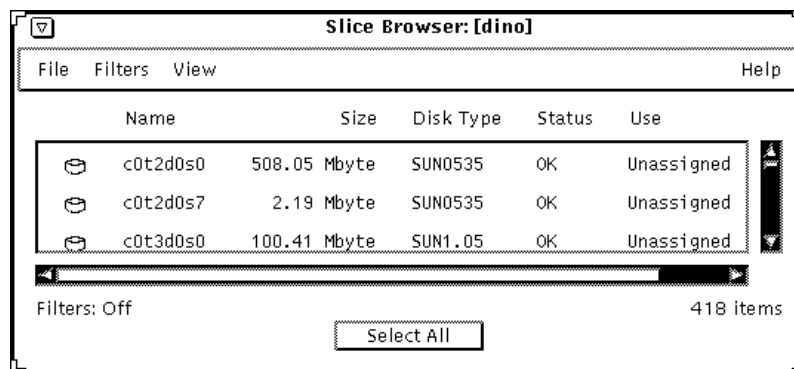
2. Open a Trans template on the Metadevice Editor canvas. Click on the Trans template.



Click here to open a Trans template.

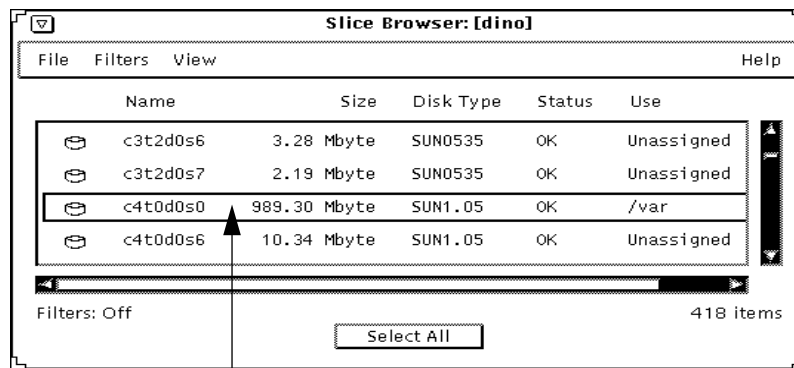
3. Open the Slice Browser.

Click on Slices on the button panel.



4. Locate the file system that will be logged.

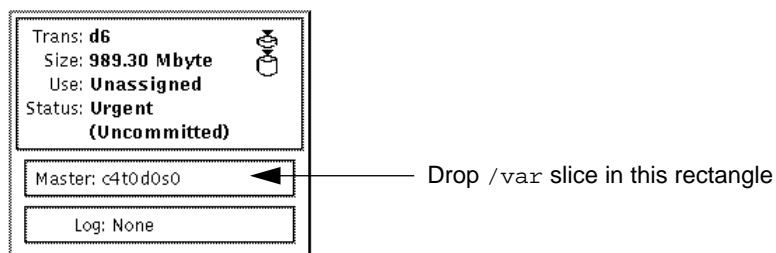
Scroll through the list of slices and locate a slice that contains the /var file system. In this example, /var is on the c4t0d0s0 slice.



Click here to select the slice.

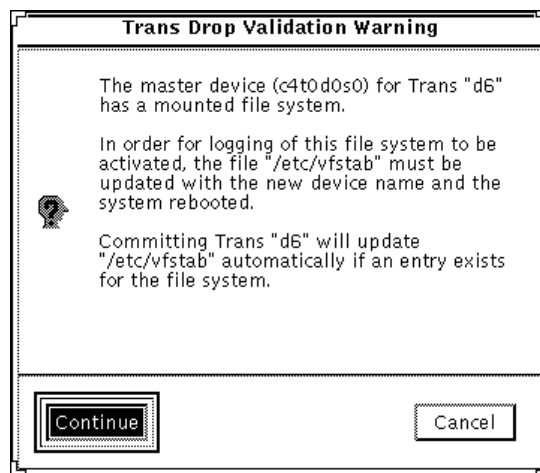
5. Move the slice to the d6 Trans device master.

Point to the c4t0d0s0 slice and press the middle button. Drag the slice to the master rectangle of the Trans device.



6. Confirm the slice will be the master.

When the c4t0d0s0 slice is dropped in the master rectangle, the validation warning dialog box is displayed. Click on the Continue button.

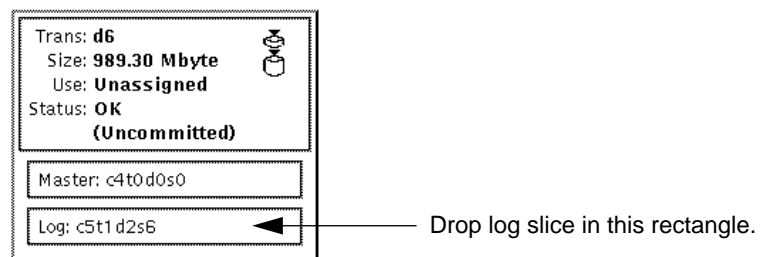


7. Select the slice that will contain the UFS log.

Scroll through the list of slices in the Slice Browser and choose one that will be the UFS log. In this example, c5t1d2s6 was selected.

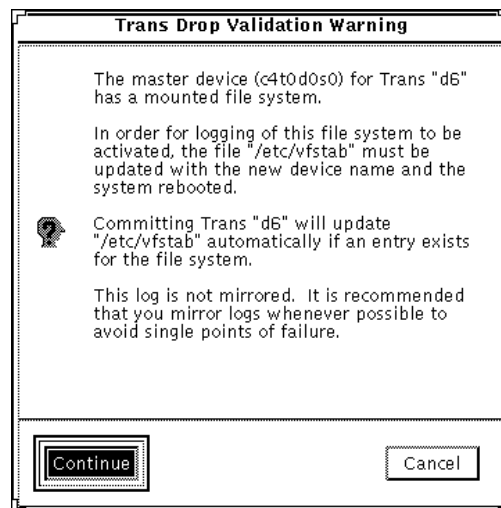
8. Drag the slice to the d6 Trans device

Point to the c5t1d2s6 slice and press and hold down the middle button. Drag the slice to the log rectangle in the d6 Trans template.



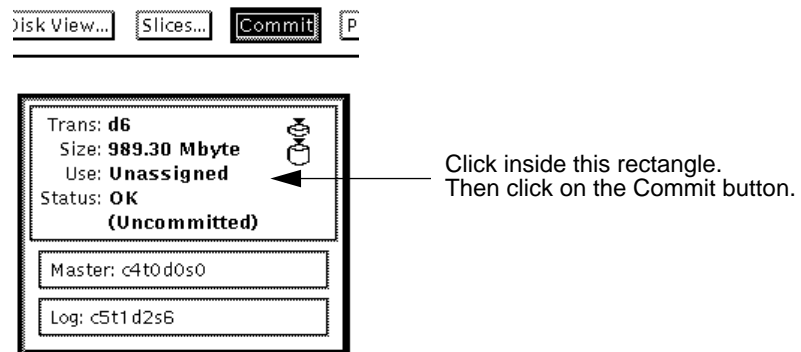
9. Confirm the slice will be the log.

When the c5t1d2s6 slice is dropped in the log rectangle, the validation warning dialog box is displayed. Click on the Continue button.



10. Commit the Trans device.

Click on the top rectangle of the Trans template. Then click on the Commit button.



11. Confirm the commit.

When you commit the d6 Trans device, a confirmation dialog box is displayed. Click on the Really Commit button.



12. Reboot the system.

Until the system is rebooted, the logging on `/var` does not begin.

Removing UFS Logging From a File Systems You Cannot Unmount

DiskSuite Tool does not support removing a UFS log from `swap`, or any other file systems (for instance, `/opt` and `/usr`) that cannot be unmounted while the system is running.

This procedure must be done using the DiskSuite command line functionality as documented in Chapter 6 of the *Solstice DiskSuite 4.0 Administration Guide*.

Using Metadevices and Mirrors as Trans Components

A logging device or a master device can be a physical component, a metadevice, a RAID or a mirror. Physical components have been used as logging and master devices in the example procedures provided in this chapter. See Chapter 4, “Concatenating and Striping,” Chapter 5, “Mirroring,” and Chapter 9, “RAID Devices” for examples of how to set up metadevices and mirrors.

To ensure data is reliable and available, use mirrors for logging devices. Otherwise, device errors on the logging device that is a physical component could cause a significant loss of the file system’s data.

The procedure you would use to create trans devices when using DiskSuite Tool would be:

- 1. Open a mirror object and populate it with two or more slices.**
- 2. Drag the mirror into the master field of the Trans device object.**

Using metadevices, for example a stripe, for logging or master devices can increase performance. Using metadevices also provides more configuration options. For example, you can expand the master device and grow its file system while the trans device is still mounted and in use. You could also expand the logging device.

Even if you are unable to mirror the logging and master devices, you should configure the logging and master devices as one-way mirrors. This enables you to add a second submirror later, without reconfiguring the Trans device.

See Chapter 5, “Mirroring,” for information on how to create mirrors.

Trans Information Window

The Trans Information window allows you to view and modify the attributes and components of a specific Trans metadvice. Figure 6-1 shows the Trans Information window. There are three ways to display the Trans Information window:

- Point to the Trans object in the Metadvice Browser and double-click. The Trans is opened on the Metadvice Editor's canvas and the Trans Information window is displayed.
- If the Trans metadvice object is on the Metadvice Editor's canvas, point inside the Trans rectangle and bring up the Trans menu. Select the Info choice.
- Double-click on a Trans object on the Metadvice Editor's canvas.

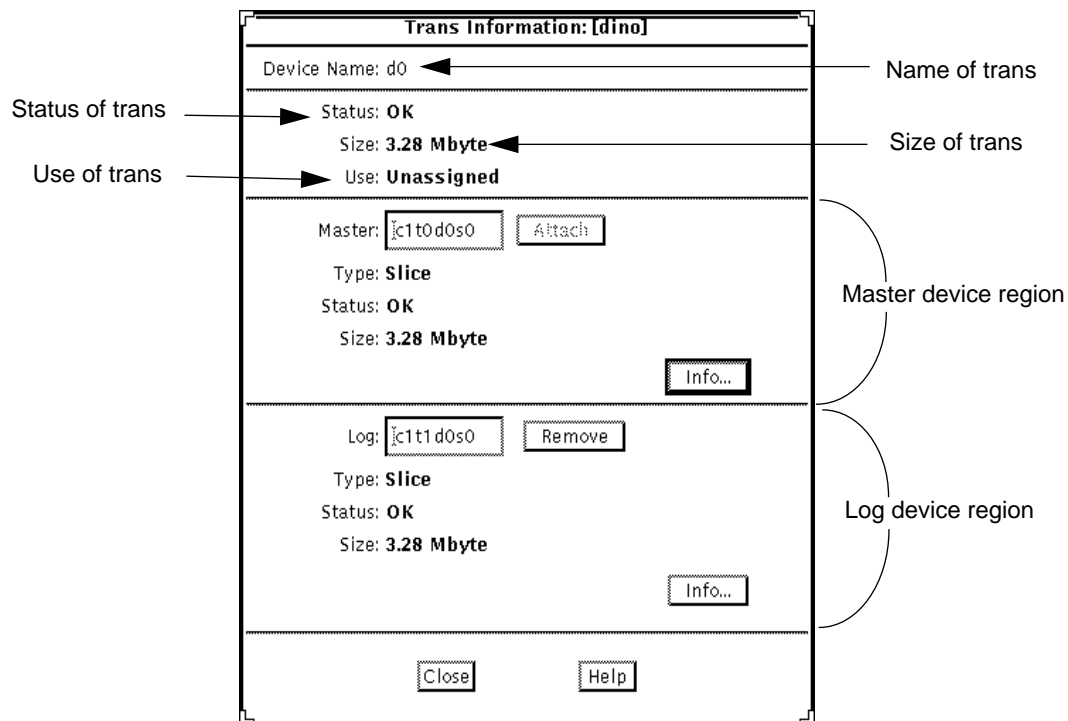


Figure 6-1 Trans Information Window

Once changes are made, the Trans object must be committed before the policy changes take effect.

Table 6-1 lists the functionality associated with the regions of the Trans Information window.

Table 6-1 Trans Information Window Functionality

Field	Functions
Device name	The metadvice name of the Trans device.
Status	Description of the Trans device status. See “Checking the Status of Trans Metadevices” on page 165 for a description of the values that can appear here.
Size	The size of the Trans device.
Use	How the Trans device is currently used (for example, file system or swap).
Master device region	<p>A region that contains the device name of the master device. The Attach button toggles between Attach and Remove. Other information in the region includes:</p> <p>Type – The type of device used as the master.</p> <p>Status – Shows the description of the Master’s status. See “Checking the Status of Trans Metadevices” on page 165 for a description of the values that can appear here.</p> <p>Size – Displays the size of the master device.</p> <p>Info – Displays the information form for the master device.</p>
Log device region	<p>A region that contains the device name where the log device is located. The Remove button toggles between Attach and Remove. Other information in the region includes:</p> <p>Type – The type of device used as the log.</p> <p>Status – Shows the description of the log’s status. See “Checking the Status of Trans Metadevices” on page 165 for a description of the values that can appear here.</p> <p>Size – Displays the size of the log device.</p> <p>Info – Displays the information form for the log device.</p>

Checking the Status of Trans Metadevices

There are three ways to check the status of Trans metadevices using DiskSuite Tool. The methods are:

- Move the Trans metadevice object from the device list and check the Status field on the object.
- Bring up the Trans Information window and check the Status field.
- Bring up the Problem List window, which is available using the Browse menu on the menu bar of the Metadevice Editor window. If a problem exists, a detailed description of the problem is given.

The Status fields of the Trans metadevice use the status keywords explained in Table 6-2.

Table 6-2 Trans Metadevice Status Keywords

Keyword	Meaning
OK	The Trans metadevice has no errors and is functioning correctly.
Detach Log (in progress)	The Trans metadevice log will be detached when the Trans metadevice is unmounted or at the next reboot of the system.
Detach Log (scheduled)	The Trans metadevice log will be detached the next time you click on the Commit button.
Attention	There is a problem with the Trans metadevice, but there is no immediate danger of losing data.
Urgent	There is a problem with the Trans metadevice and you are only one failure away from losing data. This state can only exist if the Trans metadevice contains a RAID or mirror.
Critical	Data has potentially been corrupted.

Recovering From Device Errors and File System Panics

Any device errors or file system panics you encounter must be dealt with using DiskSuite and UNIX command-line utilities. For detailed information on these commands, refer to Chapter 6 of the *Solstice DiskSuite 4.0 Administration Guide*.

Hot Spares



This chapter provides information on how to create, use, and administer hot spares by using Solstice DiskSuite Tool (`metatool(1M)`). Refer to Chapter 3, “Overview of DiskSuite Tool,” for information on using DiskSuite Tool.

Use the following table to locate specific information.

<i>Overview of Hot Spares</i>	<i>page 167</i>
<i>Hot Spare Conditions to Avoid</i>	<i>page 168</i>
<i>Defining Hot Spares</i>	<i>page 168</i>
<i>Manipulating Hot Spare Pools</i>	<i>page 175</i>
<i>Hot Spare Information Window</i>	<i>page 188</i>
<i>Checking the Status of Hot Spares</i>	<i>page 191</i>

Overview of Hot Spares

A hot spare is a component that is running (but not being used) that can be substituted for a broken component in a submirror of a two- or three-way mirror or a RAID metadevice. Failed components in a one-way mirror cannot be replaced by a hot spare, since no other copy of the data is available.

A hot spare is to a mirror or RAID metadevice what a spare tire is to a car. A spare is meant to quickly replace a flat tire, thus increasing the availability of your car. A spare tire is a temporary fix that is made with the intention that the flat tire will be repaired or replaced with a new one and the spare will be returned to the trunk.

This is exactly how hot spares should be treated. They are not intended to be used as permanent fixes when a component has failed. They are temporary fixes that can be used until a failed component is either repaired or replaced.

Components designated as hot spares cannot be used to hold replicas. They must remain ready for immediate use in the event of a component failure.

Hot Spare Conditions to Avoid

Avoid these three conditions:

- Associating hot spares of the wrong size with submirrors or RAIDs. This condition occurs when hot spare pools are defined and associated with a submirror and none of the hot spares in the hot spare pool are the same size or larger than the smallest component in the submirror. This would occur, for example, when 669-Mbyte disk drives make up the hot spare pool that is associated with a submirror that is made up of a 1-Gbyte drive.
- Having all hot spares within the hot spare pool in use. When the administrator notices that all hot spares are in use, immediate action is required. There are two possible solutions when this occurs. The first is to add additional hot spares. The second is to repair some of the components that have been replaced with hot spares. If all hot spares are in use and a submirror or RAID fails due to errors, that portion of the mirror will no longer be replicated.
- Assigning a hot spare pool to a submirror in a one-way mirror. This would provide no protection against unavailable data.

Defining Hot Spares

After a hot spare pool is defined, it can be associated with one or more submirror or RAID metadevices. Hot spare pools are named `hsp nnn` where nnn is a number in the range 000-999. A metadevice cannot be configured as a hot spare.

A hot spare can be used in one or more hot spare pools, thus allowing for the maximum amount of security from the minimum number of components. DiskSuite looks for the first available hot spare from the designated pool when errors are reported.

With DiskSuite Tool, you can also define empty hot spare pools so hot spares can be added when they become available.

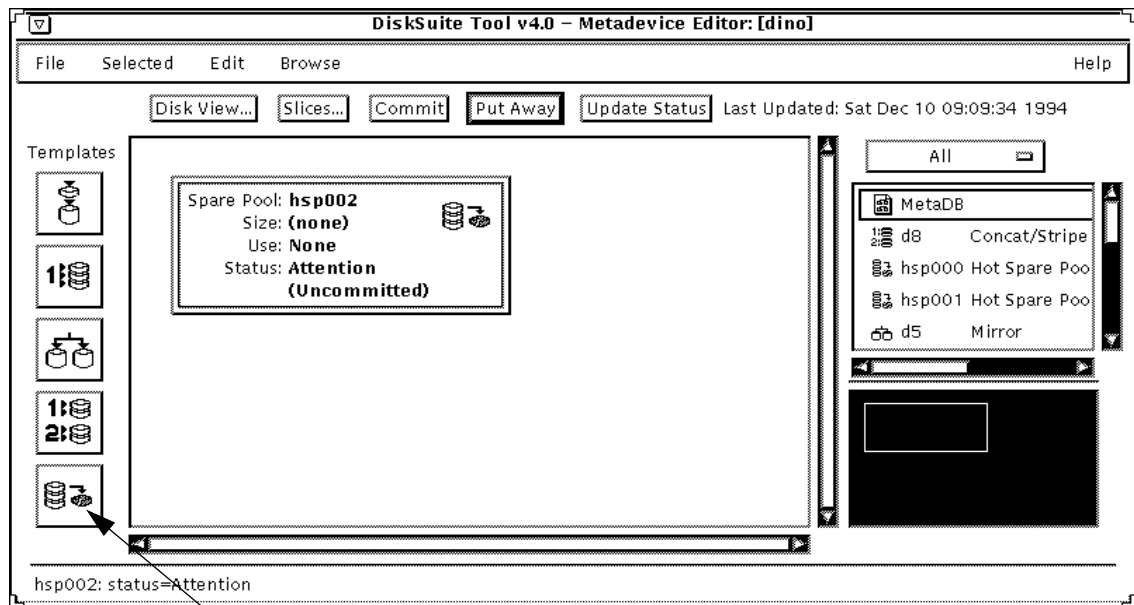
The procedure for defining hot spare pools is provided below.

1. Start DiskSuite Tool.

```
# metatool
```

2. Open a Hot Spare Pool template on the Metadevice Editor canvas.

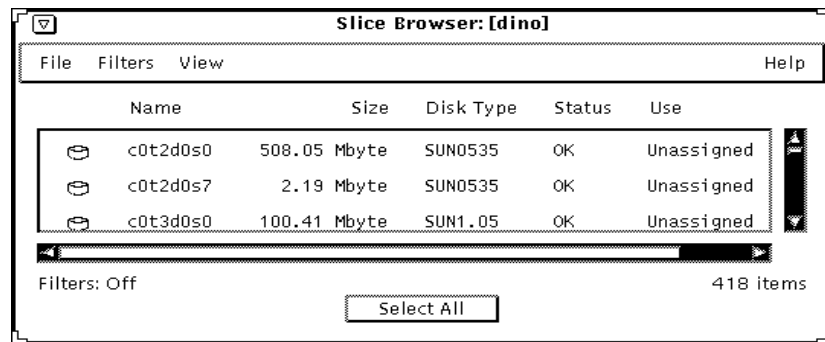
Click on the Hot Spare Pool template. An unassigned and uncommitted Hot Spare Pool template is opened on the Metadevice Editor canvas. When you point to the template, the message line displays the hot spare pool name and the status.



Click here to open the Hot Spare Pool template.

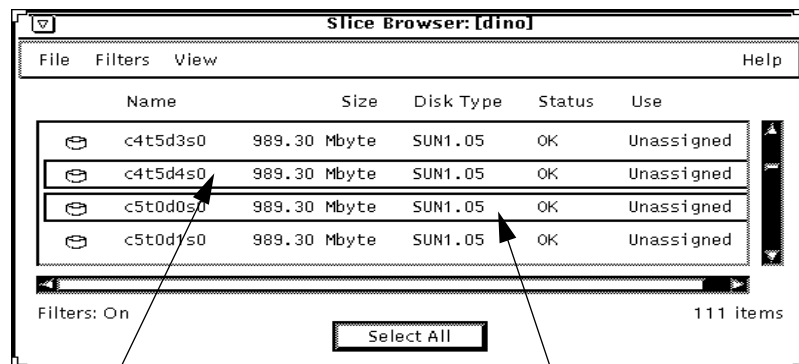
3. Open the Slice Browser.

Click on Slices on the button panel.



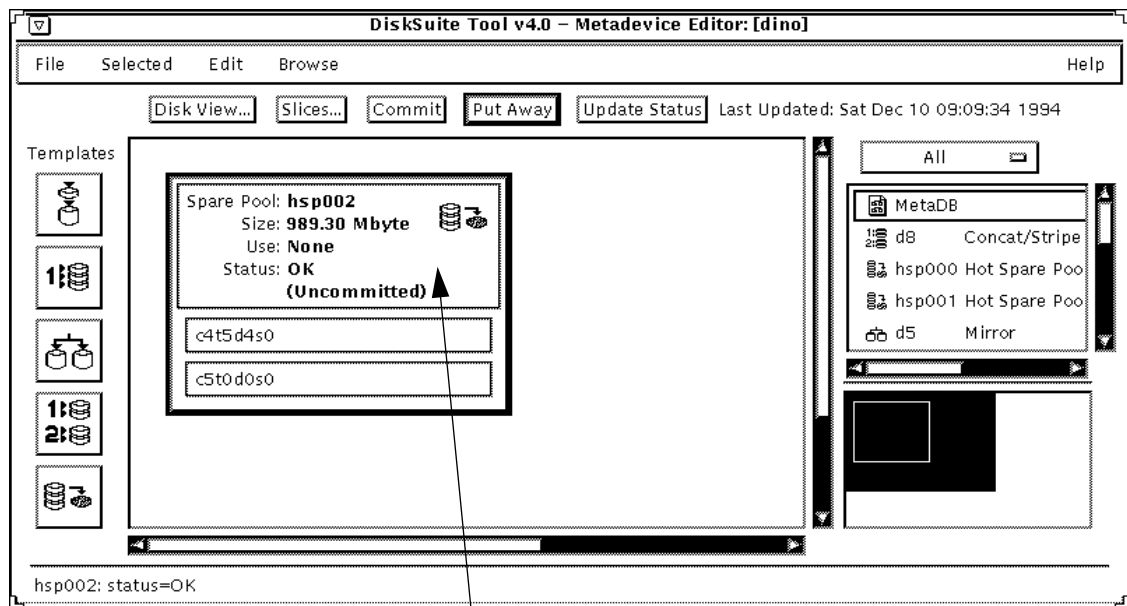
4. Select slices for the Hot Spare Pool template.

Locate the appropriate slices to be assigned to the hot spare pool. The slices should be attached to different controllers. Click on the first slice that will be assigned to the Hot Spare Pool, hold down the Control key, and click on the second slice.



5. Drag the slices into the Hot Spare Pool template.

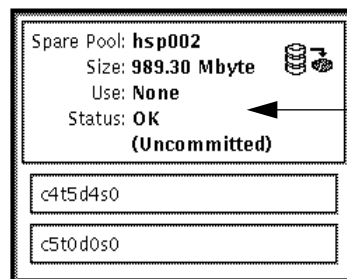
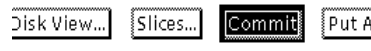
Point to one of the selected slices in the Slice Browser. Press the middle button and drag the slices to the Hot Spare Pool template.



Drop the slices inside this rectangle.

6. Commit the Hot Spare Pool template.

Click on the top rectangle of the hot spare pool template. Then click on the Commit button.



Click inside this rectangle.
Then click on the Commit button.

Associating Hot Spare Pools

After a hot spare pool has been defined it is ready to be associated with a mirror or RAID configuration. When a component failure occurs, DiskSuite searches through the list of hot spares in the assigned pool and selects the first available component that is of equal or greater disk capacity than the failed slice it will replace.

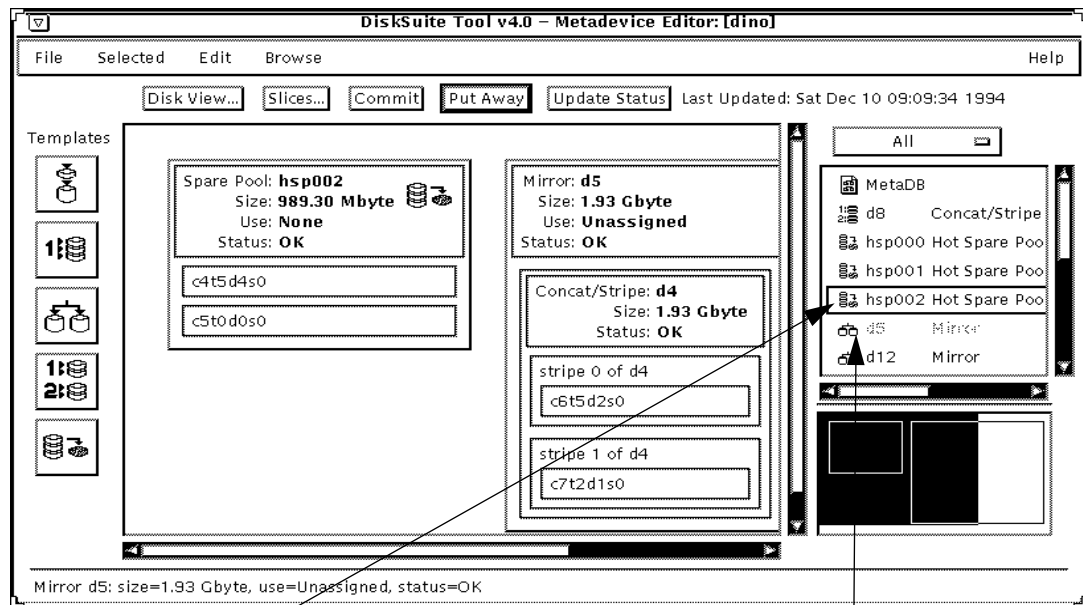
If a hot spare of adequate size is found, the hot spare's state changes to in-use and a resync operation is automatically performed. The resync operation brings the hot spare data into sync with the other submirror or the rest of the RAID metadata.

If a component of adequate size is not found in the list of hot spares, the submirror that failed is considered to be in an error condition and that portion of the submirror no longer replicates the data or, in the case of RAID, the data redundancy is no longer available.

The procedure for associating a hot spare pool with a mirror follows. The same procedure would be used for a RAID metadata.

1. Open a Mirror and Hot Spare Pool.

Scroll through the device list on the Metadevice Editor window and find the Mirror and the Hot Spare Pool. Point to each and double-click. In this example, Mirror d5 and Hot Spare Pool hsp002 are used.

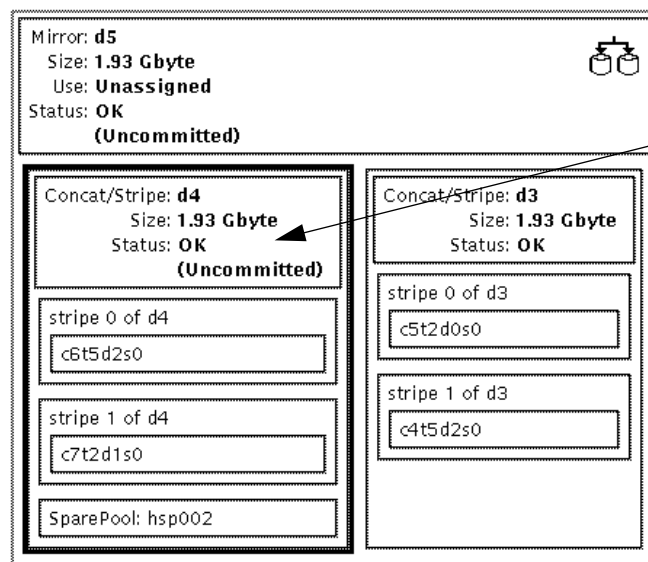


Double-click here to open the Hot Spare Pool.

Double-click here to open the Mirror.

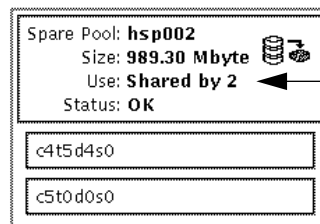
2. Drag the Hot Spare Pool template into the submirror.

Point to the upper region of the Hot Spare Pool, press the middle button and drag the object to the top of the d4 submirror.



Drop the Hot Spare Pool object into this rectangle.

The Hot Spare Pool is associated with the d3 submirror using the same procedure. After the Hot Spare Pool is associated with the second submirror, the Use field changes from None to Shared by 2.

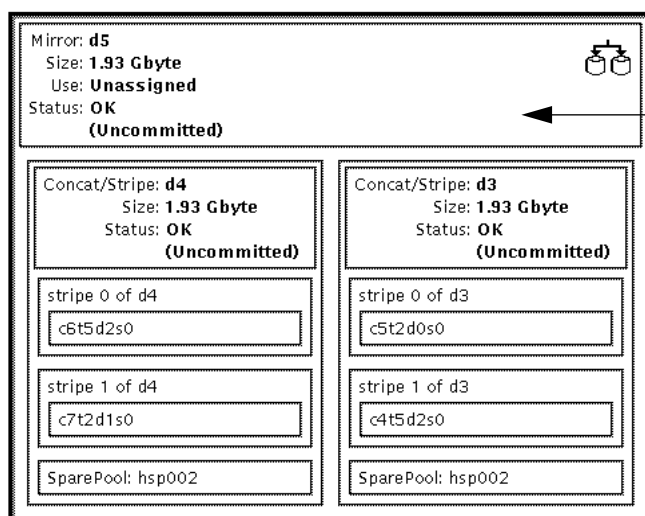


The use field shows the Hot Spare Pool is share by two metadevices.

3. Commit the mirror

Click inside the top rectangle of the d5 Mirror. Then click on the Commit button.

[Disk View...](#)
[Slices...](#)
[Commit](#)
[Put Away](#)
[Update Status](#)
 Last Updated: S:



Click inside this rectangle. Then click on the Commit button.

Manipulating Hot Spare Pools

DiskSuite Tool enables you to dynamically add, delete, replace, and enable hot spares within existing hot spare pools. You can perform the following functions on hot spare pools:

- Add hot spares to hot spare pools
- Delete hot spares from hot spare pools
- Replace hot spares in hot spare pools
- Enable hot spares
- Check the status of hot spares

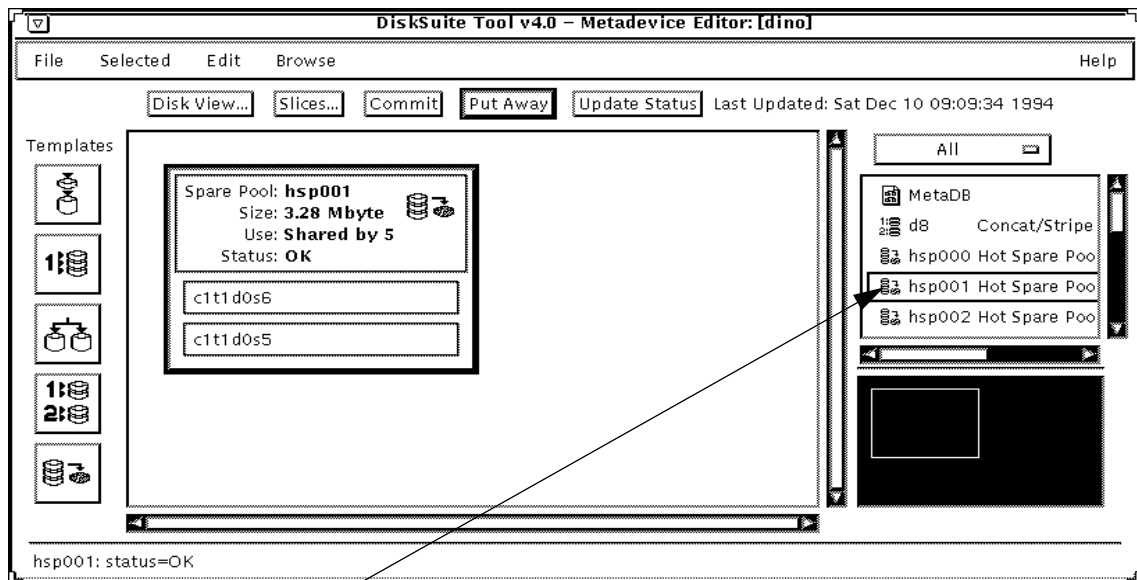
Adding Hot Spares

You can add a hot spare to one or more hot spare pools. When a hot spare is added, the existing order of the hot spares already in the pool is preserved. The new hot spare is added at the end of list of hot spares in the the hot spare pool that is specified.

The following example shows how to add a slice to a hot spare pool (hsp001).

1. Select an existing Hot Spare Pool object.

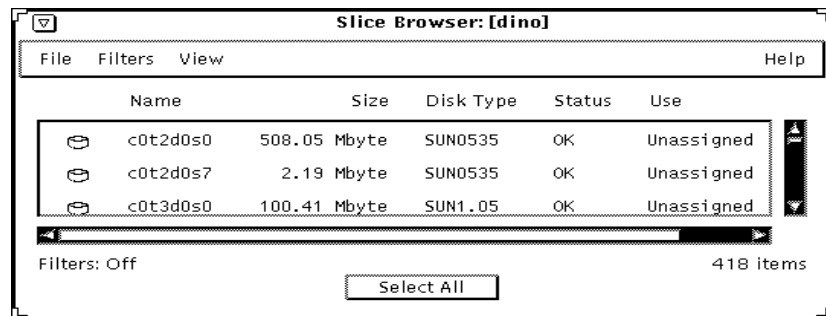
Scroll through the device list on the Metadevice Editor window and open the hot spare pool that will have the new hot spare added. In this example, Hot Spare Pool hsp001 is used.



Double-click here to open the Hot Spare Pool object.

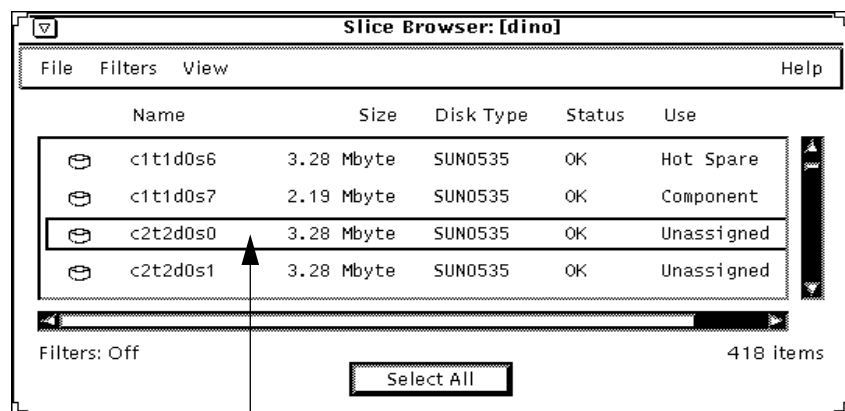
2. Open the Slice Browser.

Click on Slices on the button panel.



3. Select a slice to be added to the Hot Spare Pool template.

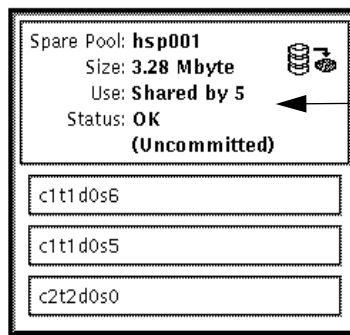
Locate a slice of the same or greater size than the slice that is already in the Hot Spare Pool. To select a slice, point to it and click. In this example, c2t2d0s0 was selected because it is the same size as the other slices in hsp001.



Click here to select the slice.

4. Drag the slice into the Hot Spare Pool object.

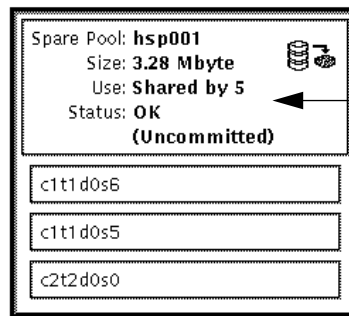
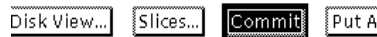
Point to the slice, press the middle button, and drag the cursor to the Hot Spare Pool.



Drop the slice in this rectangle.

5. Commit the Hot Spare Pool object.

Click on the top rectangle of the Hot Spare Pool object. Then click on the Commit button.



Click inside this rectangle.
Then click on the Commit button.

Deleting Hot Spares

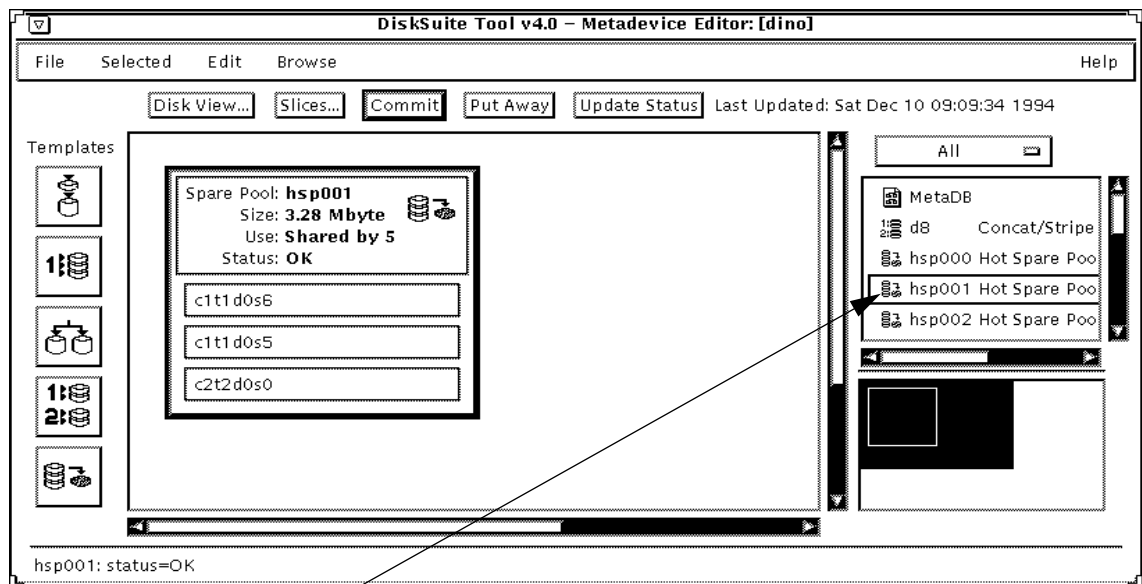
Hot spares can be deleted from any or all the hot spare pools to which they have been associated. DiskSuite Tool will not allow a hot spare to be deleted if it is in use.

When a hot spare is deleted from a hot spare pool, the positions of the remaining hot spares change to reflect the new position. For example, if the second of three hot spares in a hot spare pool is deleted, the third hot spare moves to the second position.

Use the following procedure to delete a hot spare from a hot spare pool.

1. Open the Hot Spare Pool object.

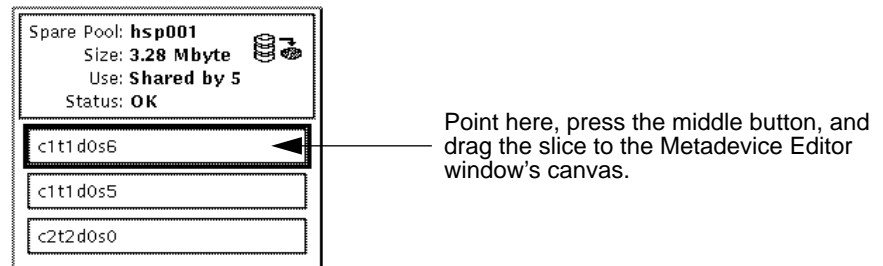
Scroll through the device list in the Metadevice Editor window and double-click on the Hot Spare Pool from which a hot spare will be deleted. In this example, Hot Spare Pool hsp001 is used.



Double-click here to open the Hot Spare Pool object.

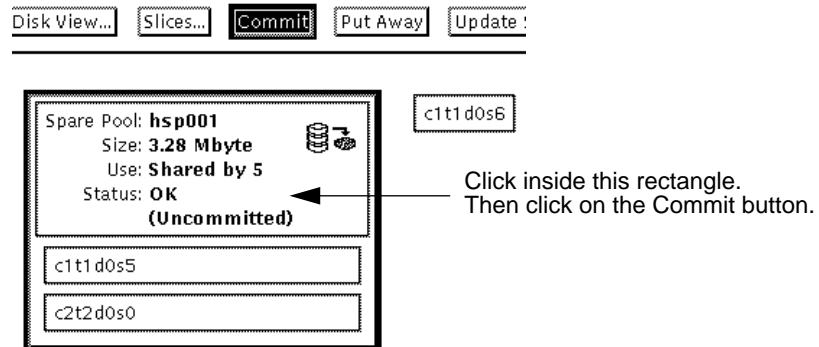
2. Remove the hot spare from the Hot Spare Pool object.

Point to the slice (hot spare) that is being removed from the Hot Spare Pool, press the middle button, and drag the slice onto the Metadevice Editor canvas. In this example, hot spare `c1t1d0s6` is being removed.



3. Commit the Spare Pool object.

Click inside the top rectangle of the Hot Spare Pool object. Then click on Commit on the button panel.



Replacing Hot Spares

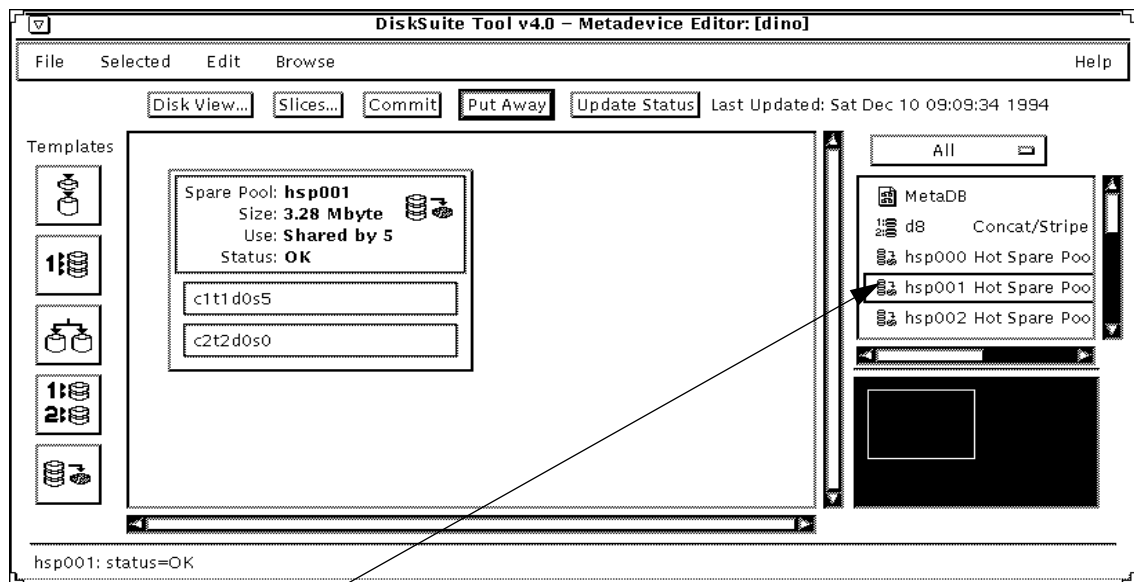
Hot spares can be replaced in any or all the hot spare pools to which they have been associated. However, hot spares that are in use cannot be replaced by other hot spares.

The order of hot spares in the hot spare pools is not changed when a replacement occurs.

Use the following procedure to replace a hot spare in a hot spare pool.

1. Open the Hot Spare Pool object.

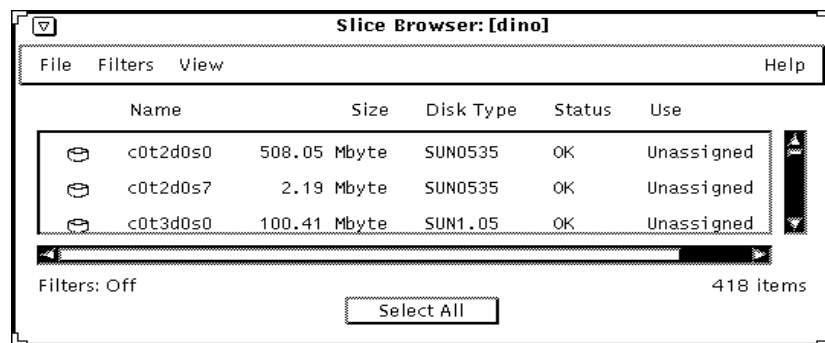
Scroll through the device list on the Metadevice Editor window and open the Hot Spare Pool object in which you want to replace a hot spare (slice). In this example, hot spare pool `hsp001` is used. The slice being replaced is `c2t2d0s0`.



Double-click here to open the Spare Pool object.

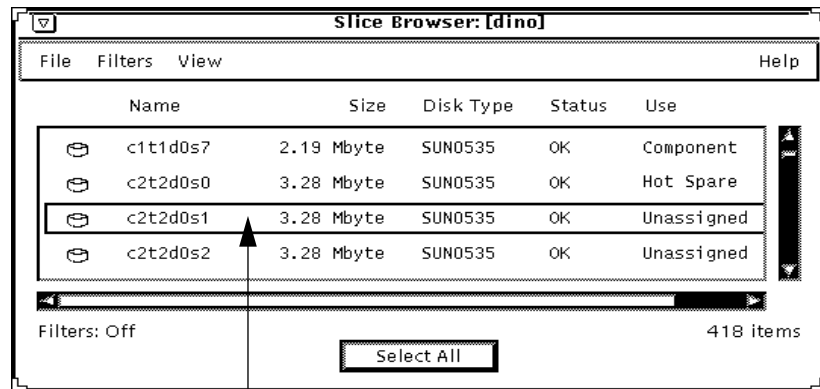
2. Open the Slice Browser.

Click on Slices on the button panel.



3. Select a replacement slice.

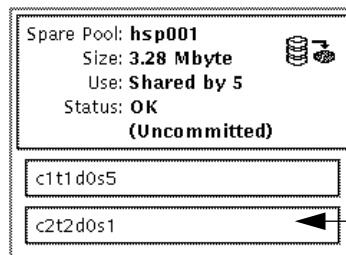
Locate an appropriate slice for replacement of slice c2t2d0s0. The replacement slice should be as large as the slice it is replacing. In this example, c2t2d0s1 is selected.



Click here to select the slice.

4. Replace the slice in the Hot Spare Pool.

Point to the replacement slice (c2t2d0s1), press the middle button, and drag the cursor to the rectangle that contains the slice to be replaced (c2t2d0s0).

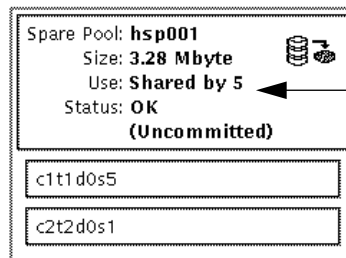


Drop the replacement slice inside this rectangle.

5. Commit the Hot Spare Pool.

Click inside the top rectangle of the Hot Spare Pool object. Then click on Commit on the button panel.

Disk View... Slices... **Commit** Put /



Click inside this rectangle.
Then click on the Commit button.

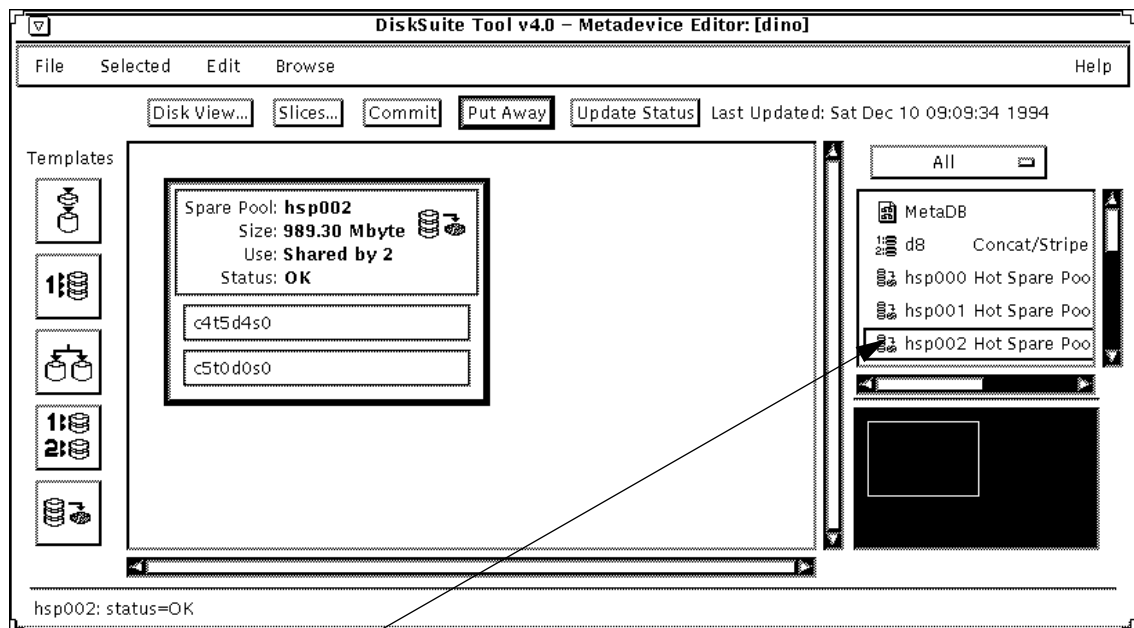
Enabling Hot Spares

When a hot spare that has been placed in the broken state is repaired, you can make it available using DiskSuite Tool.

Use the following procedure to enable a hot spare after it has been repaired.

1. Open the Hot Spare Pool object.

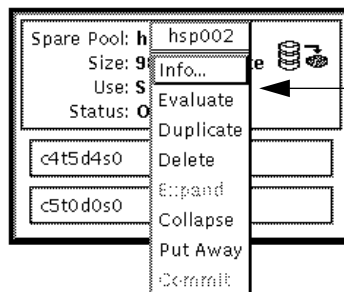
Scroll through the device list on the Metadevice Editor window and open the Hot Spare Pool object with the repaired or replaced slice. Point to the Hot Spare Pool and double-click. In this example, hot spare pool hsp002 is opened.



Double-click here to open the Hot Spare Pool object.

2. Open the Hot Spare Pool Information window.

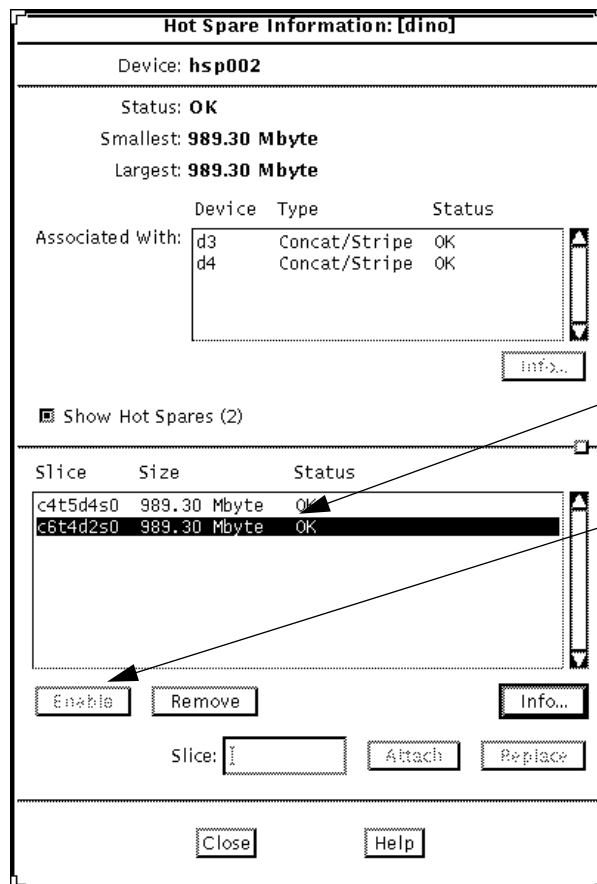
Point to the top rectangle of the Hot Spare Pool template and press the right button. Select Info from the menu.



Point inside this rectangle and press the right button. Select the Info option.

3. Select and enable the hot spare slice that has been replaced.

Click on the slice. Then click on Enable. In this example, c6t4d2s0 was selected and enabled.



Point here and click to select the slice.

Click on the Enable button.

4. Commit the Hot Spare Pool.

Click inside the top rectangle of the Hot Spare Pool object. Then click on Commit on the button panel.

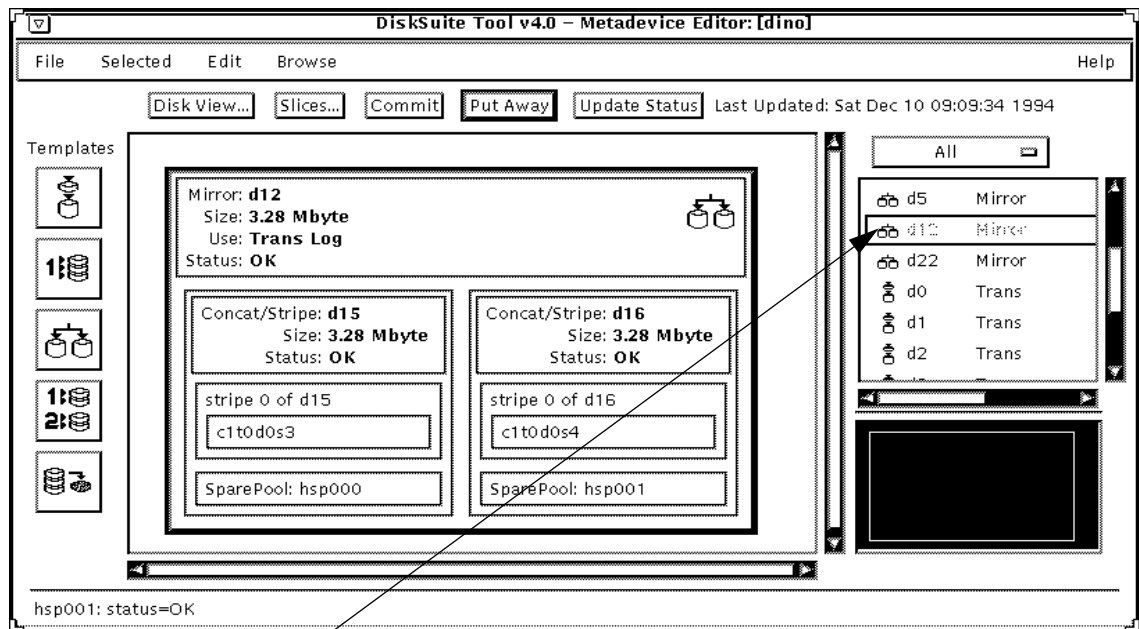
Changing the Associated Hot Spare Pool

Each hot spare pool can be associated with one or more submirrors or RAID metadevices. The association can be made and changed using DiskSuite Tool. This can be performed while the system is running, if none of the hot spares in the hot spare pool is being used by the submirror or RAID.

In the following example, the hot spare pool hsp001, is currently associated with Concat/Stripe d16 in the mirror d12. This following procedure changes the hot spare pool association to hsp000.

1. Open the d12 Mirror.

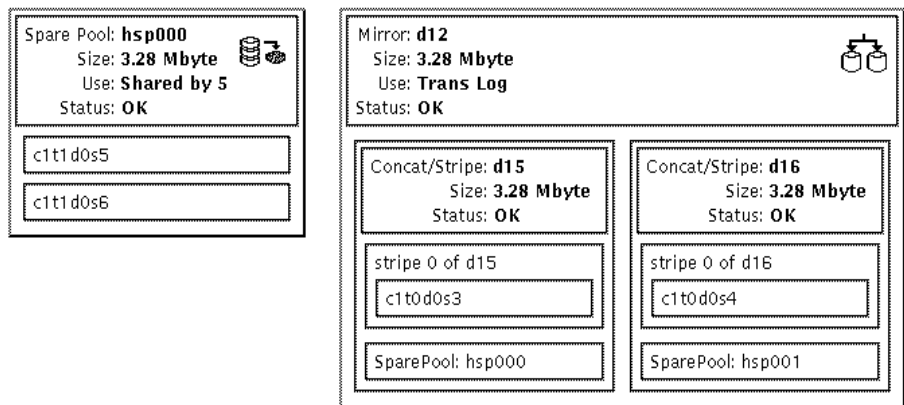
Point to the d12 Mirror in the device list and double-click.



Double-click here to open the d12 Mirror.

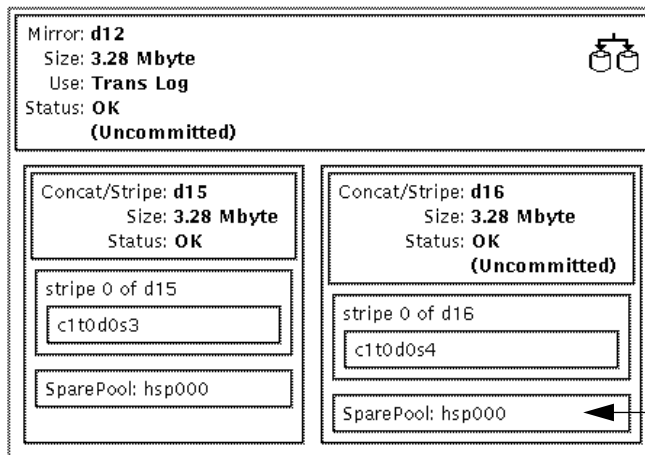
2. Open the Hot Spare Pool.

Double-click on the hsp000 Hot Spare Pool in the device list.



3. Replace hsp001 with hsp000.

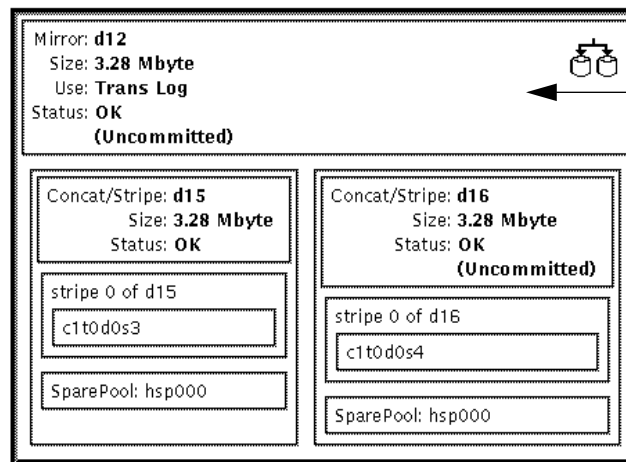
Point to hsp000, press the middle button, and drag the object to the rectangle that contains hsp001 in the d16 submirror.



When hsp000 is dropped in this rectangle, hsp001 is replaced.

4. Commit the d12 mirror.

Click inside the top rectangle of the d12 Mirror. Then click on the Commit button.



Click inside this rectangle. Then click on the Commit button.

Hot Spare Information Window

The Hot Spare Information window allows you to view and modify the attributes and components of a specific hot spare. Figure 7-1 shows the Hot Spare Information window. There are three ways to display the Hot Spare Information window:

- Point to the hot spare pool in the Hot Spare Pool Browser and double-click. The hot spare pool object is opened on the Metadevice Editor's canvas and the Hot Spare Information window is displayed.
- If the Hot Spare Pool object is on the Metadevice Editor's canvas, point inside the top of the Hot Spare Pool rectangle and bring up the Hot Spare Pool menu. Select the Info choice.

- Double-click on a Hot Spare Pool object on the Metadevice Editor's canvas.

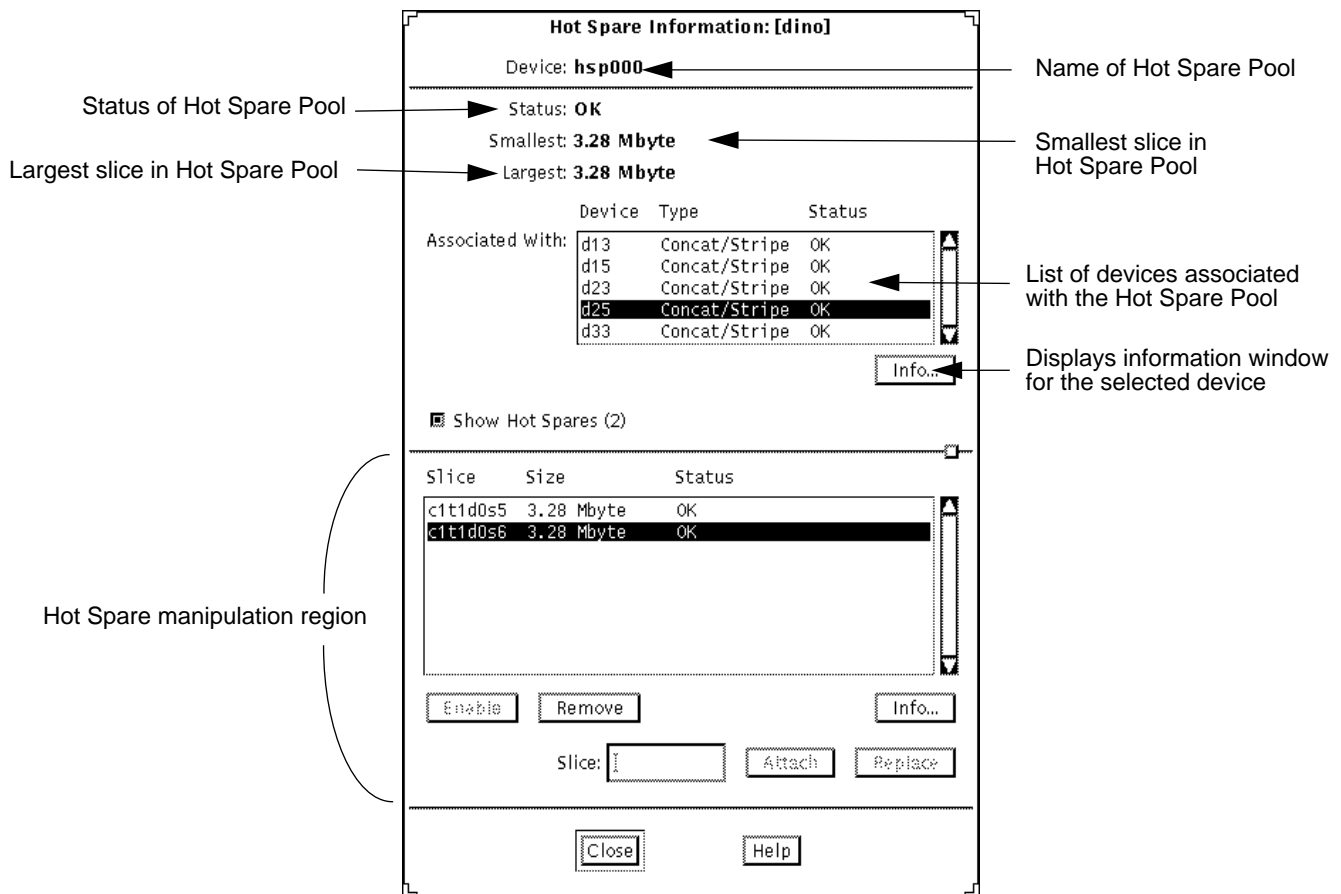


Figure 7-1 Hot Spare Information Window

The Hot Spare Pool object must be committed before the changes take effect.

Table 7-1 lists the functionality associated with the regions of the Hot Spare Pool Information window.

Table 7-1 Hot Spare Pool Information Window Functionality

Field	Functions
Device name	The name of the Hot Spare Pool.
Status	Description of the Hot Spare Pool's status. See "Checking the Status of Hot Spares" on page 191 for a description of the values that appear in this region.
Smallest	The size of the smallest slice in the Hot Spare Pool.
Largest	The size of the largest slice in the Hot Spare Pool.
Associated With	A scrolling list that displays the device names, types, and status of all metadevices associated with the Hot Spare Pool. To display information about the object either click on the object then click on Info or point to the object and double-click.
Info	Displays the Concatenation Information window for the selected (highlighted) Concat/Stripe in the Associated With region.
Hot Spare manipulation region	<p>Contains a list of all the slices included in the Hot Spare Pool. New slices can be added. Existing slices can be manipulated. The functions of the buttons include:</p> <p>Show Hot Spare – A toggle switch that shows or hides the bottom portion of the window.</p> <p>List of slices – A scrolling list of the slices included in the Hot Spare Pool.</p> <p>Enable – Enables selected slices that are disabled.</p> <p>Remove – Removes the selected slices from the Hot Spare Pool.</p> <p>Info – Displays the Slice Information window for the selected (highlighted) slice.</p> <p>Slice – Specifies a new slice to attach or replace the selected slice.</p> <p>Attach – Attaches the slice specified in the Slice field to the Hot Spare Pool. This button is active only when a slice name has been entered in the field.</p> <p>Replace – Replaces the selected spare slice with the slice entered in the field. This button is active only when a slice name has been entered in the field and a slice is selected on the list of slices.</p>

Checking the Status of Hot Spares

There are three ways to check the status of hot spare pools using DiskSuite Tool. The methods are:

- Open the hot spare pool template if it is in the device list and check the Status field.
- Bring up the Hot Spare Pool Information window and check the Status field.
- Bring up the Problem List window, which is available using the Browse menu on the menu bar of the Metadevice Editor window. If a problem exists, a detailed description of the problem is given.

The Status fields of the Hot Spare Pools use the status keywords explained in Table 7-2.

Table 7-2 Hot Spare Pool Status Keywords

Keyword	Meaning
OK	The hot spares are running and ready to accept data, but are not currently being written to or read from.
In-use	Hot spares are currently being written to and read from.
Attention	There is a problem with a hot spare or the Hot Spare Pool, but there is no immediate danger of losing data. This status is also displayed if there are no hot spares in the Hot Spare Pool or all the hot spares are in use or any are broken.

Administering Metadevices in Disksets



This chapter provides the procedure for displaying disksets using Solstice DiskSuite Tool (`metatool(1M)`). Refer to other chapters in this manual for the instructions on the manipulation of metadevices within the diskset configuration. For additional information about disksets, see Chapter 8 of the *Solstice DiskSuite 4.0 Administration Guide*. Refer to Chapter 3, “Overview of DiskSuite Tool,” for information on using DiskSuite Tool.

Use the following table to locate information specific to your task.

<i>Overview of Disksets</i>	<i>page 193</i>
<i>Administering Disksets</i>	<i>page 195</i>

Overview of Disksets

Solstice DiskSuite’s diskset feature lets you set up groups of host machines and disk drives in which all of the hosts in the set are connected to all the drives in the diskset.

DiskSuite requires that the device names be identical on each host in the diskset. There is one metadevice state database per shared diskset and one on the local diskset.

Each host in a diskset must have a local diskset that is separate from the shared diskset. The local diskset for a host consists of all drives that are not part of a shared diskset. The host’s local metadevice configuration is contained within

this local diskset in the local metadvice state database replicas. Refer to Chapter 10, “State Database Replicas,” for more information. Only the local host knows about its local diskset.

Drives in a shared diskset may not be in any other shared diskset. None of the partitions on any of the drives in a shared diskset can be mounted on, swapped on, or part of a local metadvice. Also, all of the drives in a shared diskset must be accessible by both hosts in the set.

Figure 8-1 illustrates an example of a diskset shared between two host machines, named *red* and *blue*. The shared diskset is named *relo-red*. Each host’s local diskset is shown.

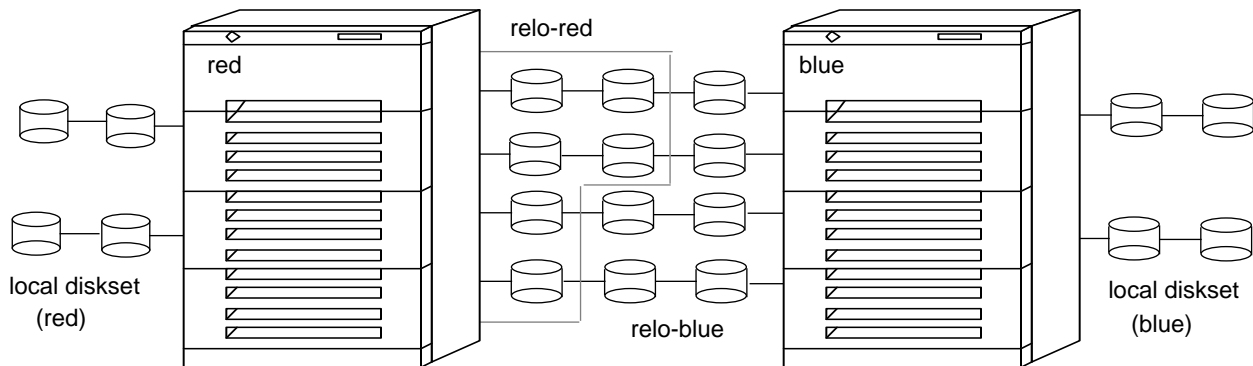


Figure 8-1 Example of a Diskset

Note – While DiskSuite’s diskset feature enables a “high-availability” configuration, it does not actually provide a high-availability environment.

Metadevices and hot spare pools in any diskset must consist of drives within that diskset. Likewise, metadevices and hot spare pools in the local diskset must be made up of drives from within the local diskset.

Administering Disksets

All the DiskSuite administration tasks described in this manual can be performed on disksets, just as they would be performed on disks in a local set. However, you can administer and create metadevices in only one diskset at a time.

Note – When disks are placed in or taken out of disksets, the actions must be done using the DiskSuite command-line interface as documented in Chapter 8 of the *Solstice DiskSuite 4.0 Administration Guide*.

To bring up DiskSuite Tool and administer a diskset, invoke the `metatool(1M)` command using the `-s` option followed by the *diskset_name*. For example:

```
# metatool -s diskset_name
```

Note – The `metatool` command must be invoked on the host that has ownership of the diskset. Once the DiskSuite Tool is running, you can perform all DiskSuite administration on the diskset, just as you would on disks not in a set. However, to administer disks in another diskset you must exit DiskSuite Tool and restart the program specifying the other *diskset_name*.

RAID Devices



This chapter provides information and procedures for creating and administering redundant arrays of inexpensive disks (RAID) metadevices using Solstice DiskSuite Tool (`metatool(1M)`). Refer to Chapter 3, “Overview of DiskSuite Tool,” for information on using DiskSuite Tool.

Use the following table to locate specific information.

<i>RAID Overview</i>	<i>page 197</i>
<i>Hardware and Software Considerations</i>	<i>page 199</i>
<i>Creating RAID Metadevices</i>	<i>page 199</i>
<i>Reconfiguring RAID Metadevices</i>	<i>page 204</i>
<i>RAID Information Window</i>	<i>page 214</i>
<i>Checking the Status of RAID Metadevices</i>	<i>page 217</i>

RAID Overview

DiskSuite supports level 5 RAID devices. A RAID configuration allows you to recover from a single disk failure. It can also be more cost effective than mirroring disks.

RAID metadevices must be comprised of three or more physical slices. Each slice is referred to as a component. A RAID metadevice can be grown by concatenating additional slices to the metadevice.

RAID level 5 includes multiple physical slices used to simulate a single large slice. A single sector on one of these physical slices contains either a sector's worth of data, or parity information relating to the data on the same sector of all other slices in the array.

In order to eliminate a parity slice as a bottleneck, no one physical slice will hold all of the parity information; it will be placed on different slices for different sectors.



Caution – Do not attempt to create a RAID device with slices that contain an existing file system. The data on the slices will be erased when the RAID goes through a resync.

The advantages of a RAID level 5 configuration are:

- It can recover from a disk failure.
- It is more cost effective than mirroring disks.

DiskSuite Tool automatically resyncs RAID metadevices when components are replaced or new components are added.

Operation of RAID

The following operations are supported for RAID metadevices using DiskSuite Tool:

- Configuring and defining RAID metadevices
- Concatenating new components to existing RAID metadevices
- Allocating hot spare pools to RAID metadevices to provide component backup in the event of original component failures
- Replacing failed components or in-use hot spare components with a new component
- Viewing the status of RAID metadevices
- Clearing RAID metadevices

Each of these operations is discussed in more detail in the following sections.

Hardware and Software Considerations

There are both hardware and software considerations that affect RAID metadevices. For a detailed discussion of all hardware and software considerations, refer to Chapter 9 of the *Solstice DiskSuite 4.0 Administration Guide*.

Creating RAID Metadevices

This subsection describes how to create a RAID metadevice using DiskSuite Tool. In the example used, the RAID metadevice will contain three slices of equal size. An interlace value of 32 Kbytes will be set.

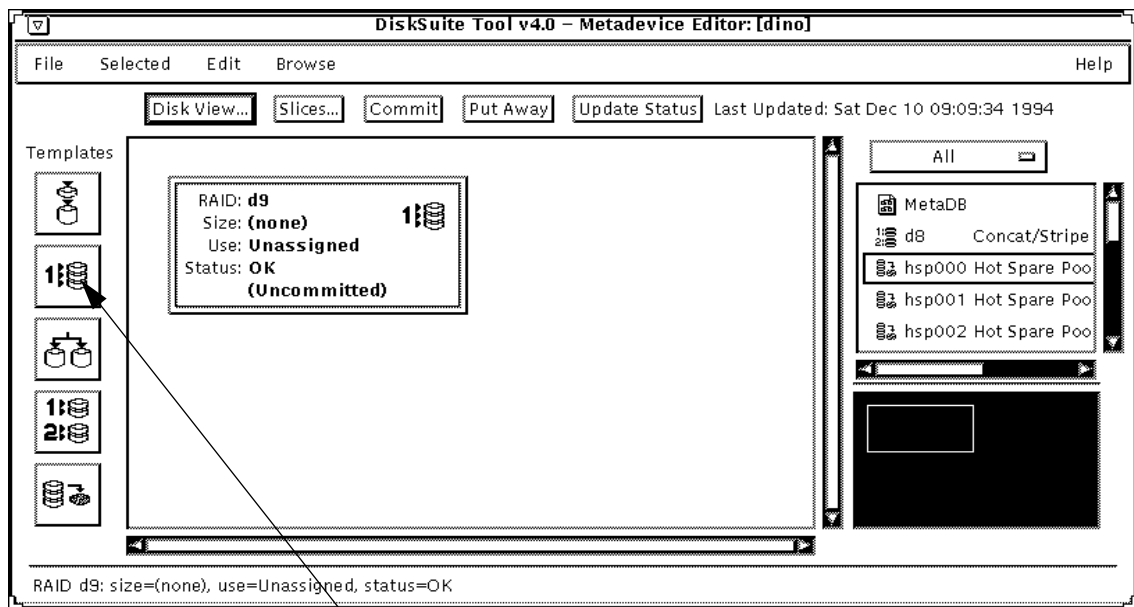
The following example shows how to create a RAID metadevice.

1. Start DiskSuite Tool.

```
# metatool
```

2. Open a RAID template on the Metadevice Editor canvas.

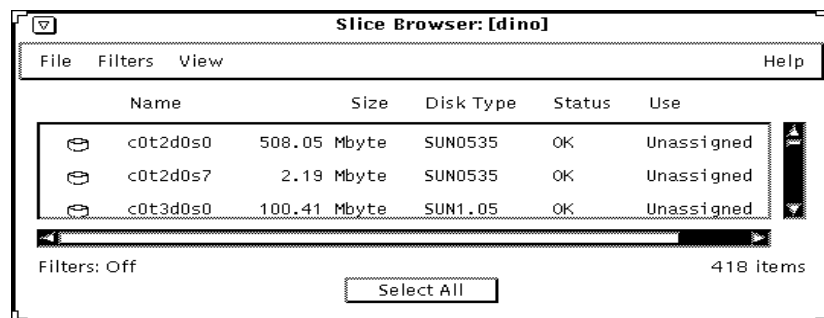
Click on the RAID template. An unassigned and uncommitted RAID template is opened on the Metadevice Editor canvas. When you point to the template, the message line displays the metadevice name, size, use, and status.



Click here to open a RAID template

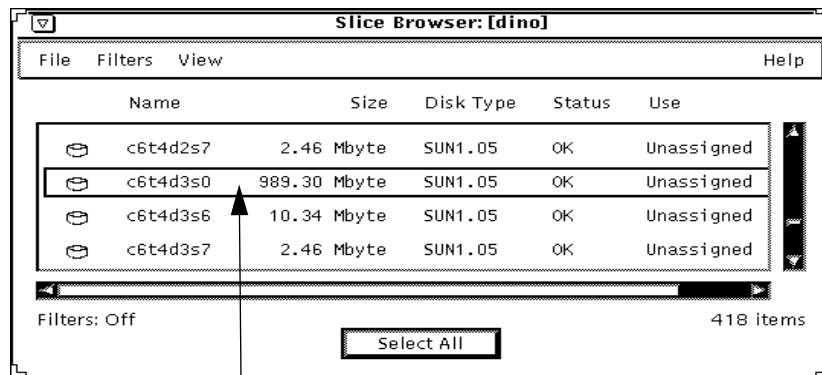
3. Open the Slice Browser.

Click on Slices on the button panel.



4. Select the first slice for the RAID device.

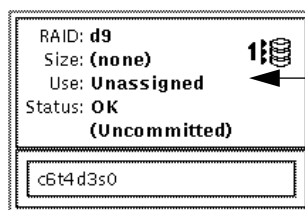
Scroll through the list of slices and select the first slice for the RAID device.



Click here to select the slice.

5. Drag the slice into the RAID template.

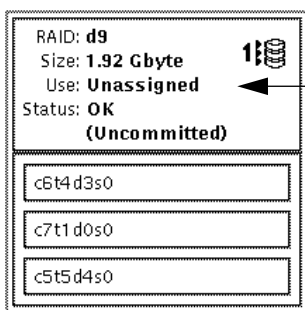
Point to the slice, press the middle button, and drag the cursor to the RAID template. The size is represented as none, because the size is not known until there are at least three slices in the RAID metadvice.



Drop the slice in this rectangle.

6. Drag additional slices into the RAID template.

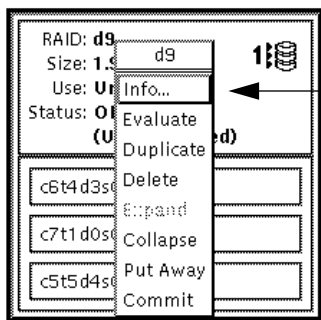
Scroll through the Slice Browser and locate additional slices for the RAID device. The slices should be the same size as the initial slice and ideally attached to different controllers. In this example, slice `c6t4d3s0` was the first slice. The second and third slices are `c7t1d0s0` and `c5t5d4s0`. All the slices are 989 Mbytes. However, the Size field of the RAID template reports the size as 1.92 Gbytes because the equivalent capacity of one disk is used for parity.



Drop the additional slices in this rectangle.

7. Optionally change the interlace size.

By default the interlace value of the RAID is set to 16 Kbytes. To change the interlace value, point to the top of the RAID template rectangle, press the right button, and select Info from the menu.



Point inside this rectangle and press the right button. Select Info on the menu.

The RAID Information window is displayed, showing information about the RAID device. To change the interlace value, point to the Custom button and click. Enter the new value and click on Attach.

For more about the RAID Information window, see page 214.

RAID Information: [dino]

Device Name:

Status: **OK**
 Size: **1.93 Gbyte**
 Use: **Unassigned**

Hot Spare Pool:

Interlace:

☒ Default 16 Kbytes
☐ Custom 32 Kbytes

☒ Show Slices (3)

Slice	Size	Replicas	Status
c4t0d2s0	989.30 Mbyte	0	OK
c5t0d0s0	989.30 Mbyte	0	OK
c6t0d4s0	989.30 Mbyte	0	OK

Slice:

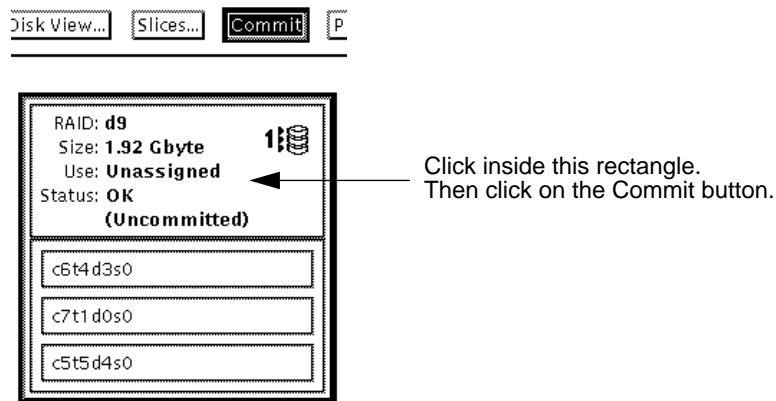
Default interlace value is set to 16 Kbytes.

Click on Custom to change the value. The value entered is 32 Kbytes.

Point to the Attach button and click to assign the new interlace value.

8. Commit the RAID metadvice.

Click inside the top rectangle of the d9 RAID metadvice. Then click on Commit on the button panel.



Reconfiguring RAID Metadevices

Reconfiguration of a RAID metadvice includes the following operations:

- Concatenating additional components to the RAID metadvice
- Assigning a hot spare pool to the metadvice to provide a backup in the event that errors are encountered on any of the components of the metadvice
- Replacing components (error or non-errored)

Instructions for performing each of these reconfiguration options are provided in the following subsections.

Concatenating Components

You can concatenate a component to a RAID metadvice to expand the device's capacity.

When a slice is added to the RAID metadvice, the slice becomes a concatenation to the initial metadvice. The new slice will not contain RAID level 5 parity information. But the data on the new slice will be protected by the parity that already existed in the metadvice.

Note – After the slice is added and the RAID metadvice is committed, the slice cannot be removed. However, the slice can be replaced by a slice of equal or greater size.

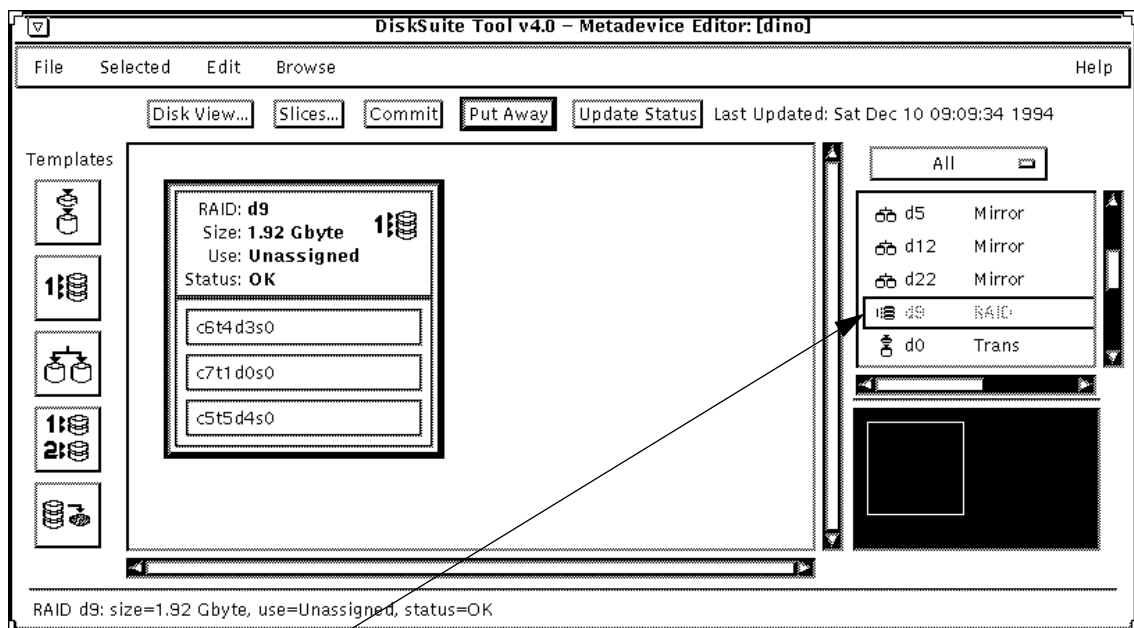
The following example shows how to add a slice to an existing RAID metadvice. The added slice will be a single concatenated component.

1. Start DiskSuite Tool.

```
# metatool
```

2. Open the RAID metadvice object on the Metadvice Editor canvas.

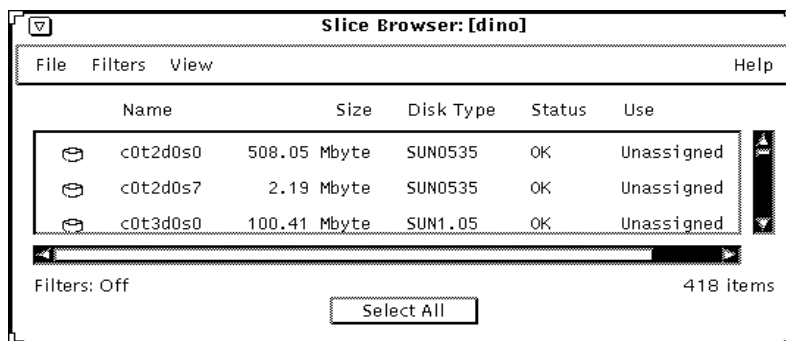
Scroll through the device list on the Metadvice Editor to find the RAID metadvice object that will receive an additional slice. Point to the device and double-click the left button.



Double-click here to open the RAID metadvice.

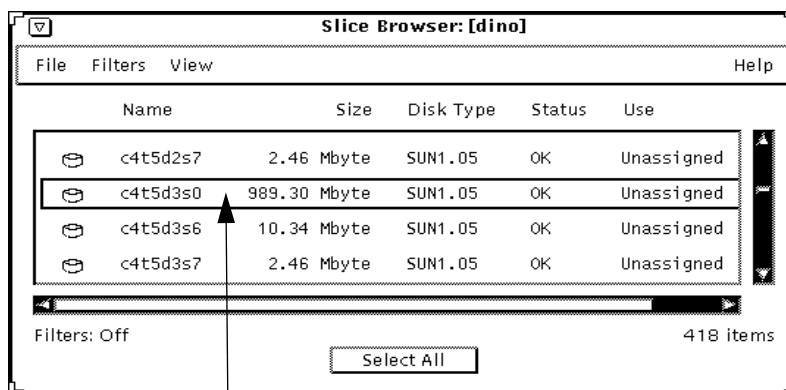
3. Open the Slice Browser.

Click on Slices on the button panel.



4. Select a slice to be added to the RAID metadvice.

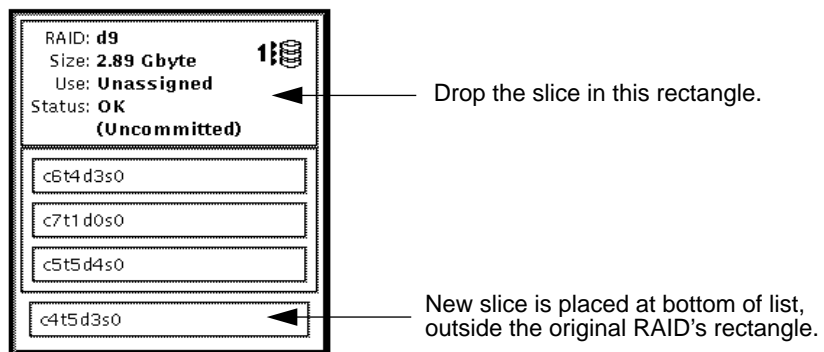
Scroll through the list of slices and select a slice that will be added to the metadvice. The slice must be at least as large as the smallest slice in the metadvice. In this example, c4t5d3s0 was selected because it is the same size as the other slices and is connected to a different controller.



Click here to select the slice.

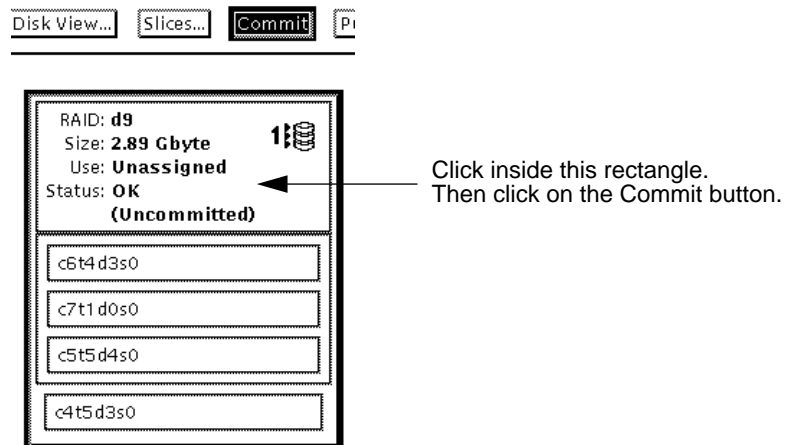
5. Drag the slice into the RAID metadvice.

Point to the `c4t5d3s0` slice, press the middle button, and drag the slice to the top rectangle of the RAID metadvice.



6. Commit the RAID metadvice.

Click on the top rectangle of the `d9` RAID metadvice. Then click on Commit on the button panel.



Assigning a Hot Spare Pool

Hot spare pools may be allocated, deallocated, or reassigned at any time unless a component in the hot spare pool is being used to replace an errored component in the RAID metadvice.

After the hot spare pool has been assigned to the metadvice, any failed components will be replaced by the first available appropriately sized component in the hot spare pool.

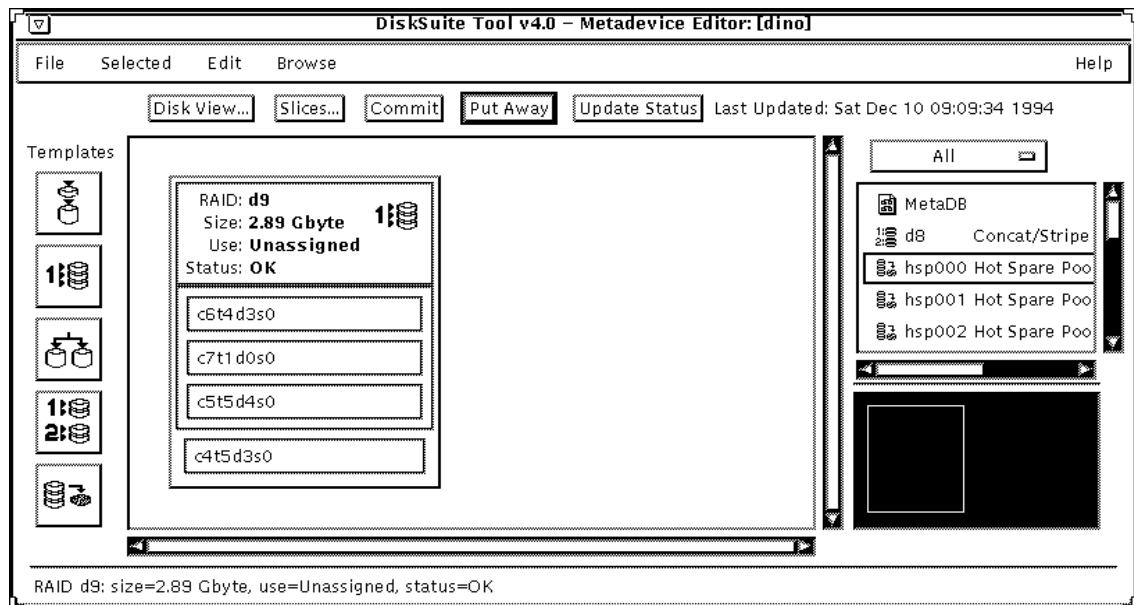
The following steps describe how to assign a hot spare pool to a RAID metadvice.

1. Start DiskSuite Tool.

```
# metatool
```

2. Open the d9 RAID metadvice object on the Metadvice Editor canvas.

Scroll through the device list on the Metadvice Editor to find the RAID metadvice object that will be associated with a hot spare pool. Once located, point to the device and double-click the left button.



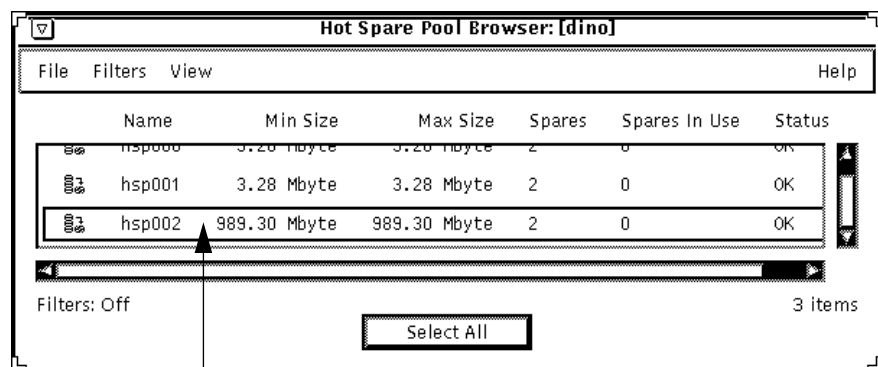
3. Open the Hot Spare Pool Browser.

Point to the Browse menu on the menu bar of the Metadevice Editor window, press the right button, and select Hot Spare Pools.



4. Select a hot spare pool object.

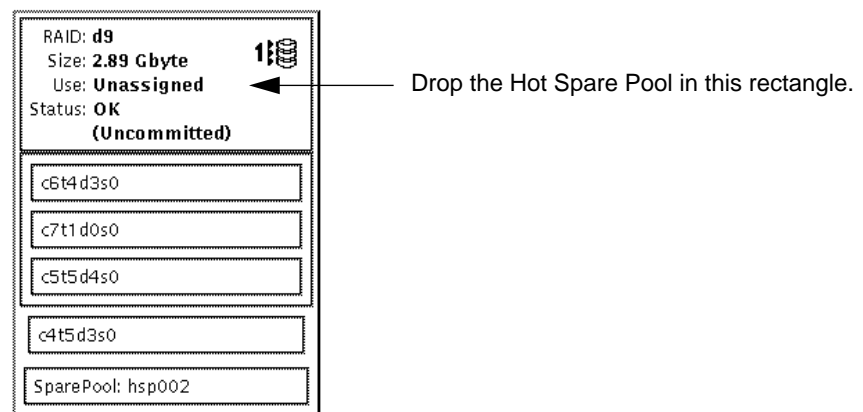
Scroll through the list of Hot Spare Pool objects and select the one that will be assigned to the RAID metadevice. The hot spare pool must have slices the same size or larger than the slices in the RAID metadevice. Ideally, the Hot Spare Pool will contain hot spares that are on a different controller than the slices in the RAID metadevice.



Click here to select the slice.

5. Assign the Hot Spare Pool object to the RAID metadata.

Point to the Hot Spare Pool object, press the middle button, and drag the object to the top rectangle of the RAID metadata.



6. Commit the RAID metadata.

Click inside the top rectangle of the RAID metadata. Then click on Commit on the button panel.

Replacing Components

When an I/O error is detected on a component of a RAID metadata, no further actions will be performed on that component. You should replace the component or perform some other type of error recovery to return the state of the RAID metadata to non-errored. This reduces the possibility of lost data.



Caution – Replacing an errored component when multiple components are in error may cause data to be fabricated. The integrity of the data in this instance is questionable.

The following example shows how to replace an errored slice in a RAID metadata.

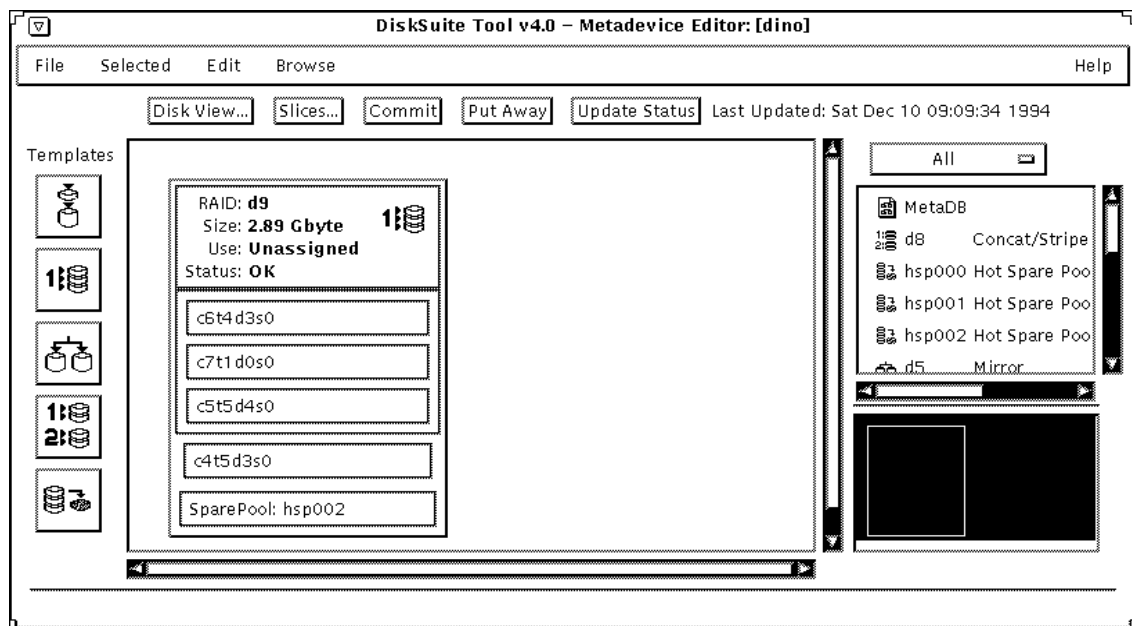
During component replacement, data for the errored component will be recovered from a hot spare component or from the available, non-errored, components. This is possible due to the parity segments that are generated during normal I/O operation.

The following steps show how to replace a component in a RAID metadvice.

1. Start DiskSuite Tool.

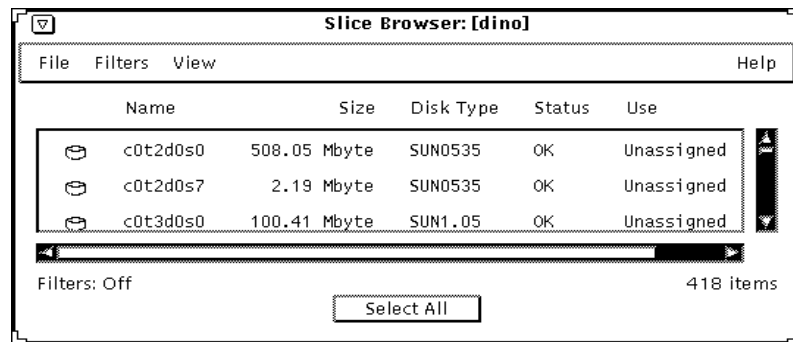
```
# metatool
```

- 2. Open the RAID metadvice object on the Metadvice Editor canvas.**
 Scroll through the device list on the Metadvice Editor and find the RAID metadvice object that will receive a replacement slice. Point to the RAID metadvice and double-click the left button.



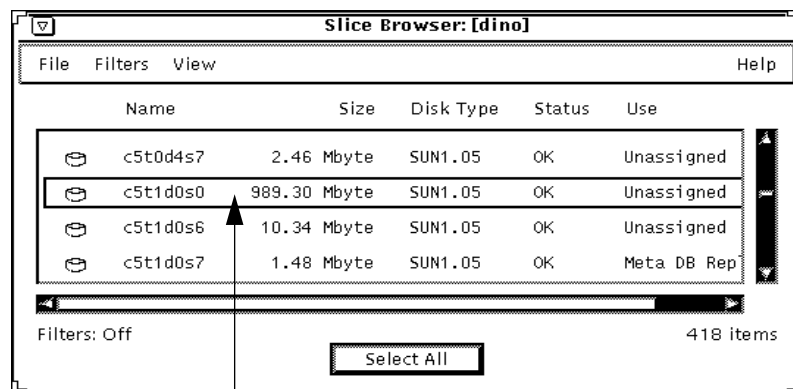
3. Open the Slice Browser.

Click on Slices on the button panel.



4. Select a slice to replace a slice in the RAID metadvice.

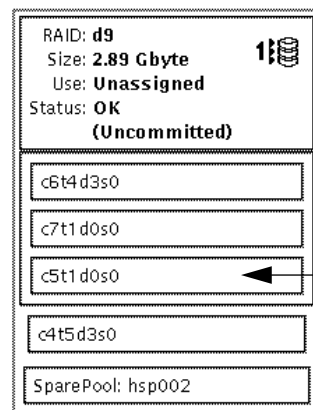
Scroll through the list of slices and select a slice that is at least as large as the smallest component in the RAID metadvice. In this example, c5t1d0s0 was selected. It will replace slice c5t5d4s0 in the RAID metadvice.



Click here to select the slice.

5. Replace the slice in the RAID metadvice.

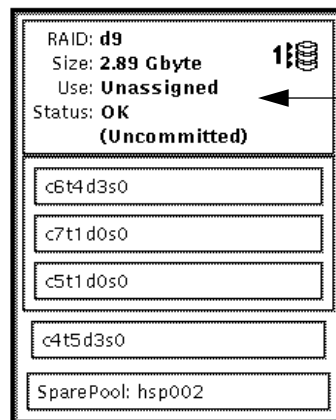
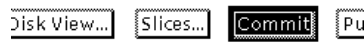
Point to the selected slice in the Slice Browser, press the middle button, and drag the slice to the rectangle that contains the slice that will be replaced.



Drop the replacement slice inside the rectangle that contains the slice that is being replaced.

6. Commit the RAID metadvice.

Click inside the top rectangle of the RAID metadvice. Then click on Commit on the button panel.



Click inside this rectangle. Then click on the Commit button.

RAID Information Window

The RAID Information window allows you to view and modify the attributes and components of a specific RAID metadvice. Figure 9-1 shows the RAID Information window. There are three ways to display this window:

- Point to the RAID device in the Metadvice Browser and double-click the left button. The RAID device is opened on the Metadvice Editor's canvas and the RAID Information window is displayed.
- If the RAID metadvice is on the Metadvice Editor's canvas, point inside the top of the rectangle and bring up the RAID menu. Select the Info choice.
- Double-click on a RAID metadvice on the Metadvice Editor's canvas.

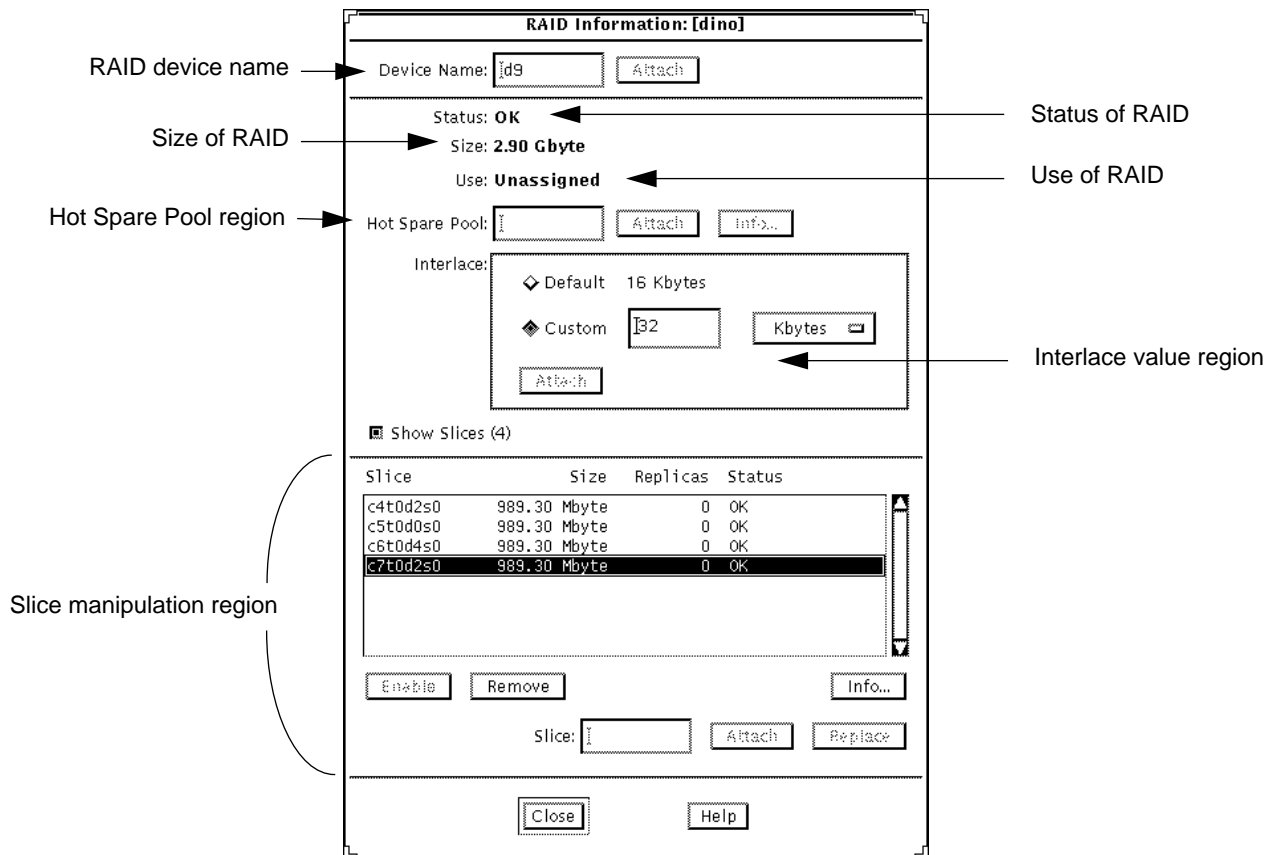


Figure 9-1 RAID Information Window

Once changes are made, the RAID metadvice must be committed before the changes take effect.

Table 9-1 lists the functionality associated with the regions of the RAID Information window.

Table 9-1 RAID Information Window Functionality

Field	Functions
Device name	The metadvice name of the RAID.
Status	Description of the RAID's status. See "Checking the Status of RAID Metadevices" on page 217" for a description of the values that appear in this region.
Size	The size of the usable disk space. This does not include the size set aside for parity.
Use	The use of the RAID, for example, file system or swap. If the use of the RAID is a Trans Log, a Show Trans button is positioned to the right of the field.

Table 9-1 RAID Information Window Functionality

Field	Functions
Hot Spare Pool	<p>This field enables assigning a Hot Spare Pool to the RAID metadvice. It has the following functions:</p> <p>Attach/Detach – Attaches or detaches the specified Hot Spare Pool to the RAID metadvice.</p> <p>Info – Displays the Hot Spare Pool Information window for the specified Hot Spare Pool.</p>
Interlace value region	<p>The default interlace value is 16 Kbytes. To change the interlace value, click on the Custom button and enter the value in the field. The menu button to the right of the field enables you to specify the units used. The values on the menu are Gbytes, Mbytes, Kbytes, and Sectors. The default is Kbytes. After the Custom field is complete, the Attach button is used to assign the interlace value to the RAID. After a RAID is committed, the interlace value cannot be changed.</p>
Slice manipulation region	<p>The following functionality is available in this region:</p> <p>Show Slices – A toggle switch that shows or hides the scrolling list of components at the bottom of the window.</p> <p>Scrolling List – A list of the slices included in the RAID metadvice. The information in this region includes the name of the slice, size, number of state database replicas on the slice and the status.</p> <p>Remove – Removes the selected slices.</p> <p>Enable – Activates the selected slices if they are disabled.</p> <p>Slice – Enables you to specify a new slice to attach to the RAID or replace the selected slice.</p> <p>Attach – Attaches the slice specified in the Slice field to the RAID metadvice. This button is active only when a slice name is entered in the field.</p> <p>Replace – Replaces the selected RAID slice with the slice entered in the Slice field. This button is active only when a slice name has been entered in the field and a slice is selected from the scrolling list.</p> <p>Info – Displays the Slice Information window for the selected (highlighted) slice.</p>

Checking the Status of RAID Metadevices

There are three ways to check the status of RAID metadevices using DiskSuite Tool. The methods are:

- Move the RAID metadevice object from the device list and check the Status field on the object.
- Bring up the RAID metadevice information window and check the Status field.
- Bring up the Problem List window, which is available using the Browse menu on the menu bar of the Metadevice Editor window. If a problem exists, a detailed description of the problem is given.

The Status fields of the RAID metadevice use the status keywords explained in Table 9-2.

Table 9-2 RAID Status Keywords

Keyword	Meaning
OK	The RAID metadevice has no errors and is functioning correctly.
Attention	There is a problem with the RAID metadevice, but there is no immediate danger of losing data.
Urgent	There is a problem with the RAID metadevice and you are only one failure away from losing data.
Critical	Data has potentially been corrupted.

State Database Replicas

10 

This chapter provides information on how to set up and administer the state database and its associated replicas with the Solstice DiskSuite Tool (`metatool(1M)`). Refer to Chapter 3, “Overview of DiskSuite Tool,” for information on using DiskSuite Tool.

Use the following table to locate specific information.

<i>Overview of the State Database Replicas</i>	<i>page 219</i>
<i>Planning Locations of Replicas</i>	<i>page 221</i>
<i>Creating the Initial State Database Replicas</i>	<i>page 222</i>
<i>Adding Replicas</i>	<i>page 225</i>
<i>Removing Replicas</i>	<i>page 228</i>
<i>Metadevice State Database Information Window</i>	<i>page 231</i>
<i>Checking the Status of Replicas</i>	<i>page 234</i>

Overview of the State Database Replicas

A state database replica is a dedicated portion of a disk. It is similar to a disk label. The space occupied by the replica is reserved for the exclusive use of the metadevice state database; it cannot be used for any other purpose.

State database replicas are critical to the operation of all metadevices because they provide a memory service for DiskSuite. The replicas keep track of configuration and status information for all metadevices, mirrors, trans devices, RAID devices, and hot spares. The replicas also keep track of error conditions that have occurred.

The replicated state database information keeps DiskSuite operating. Without replicas (copies) of the same information for comparison, DiskSuite does not know the current running state of metadevices. This chapter, along with Chapter 10 of the *Solstice DiskSuite 4.0 Administration Guide* and the `metadb(1M)` manual page provide a detailed discussion of how the replicas are used by the metadisk driver.

Each replica can exist on either a dedicated disk partition or on space reserved for a replica within a striped or concatenated metadvice or a logging device. You can store multiple replicas in a single disk partition. However, placing multiple replicas on a single disk reduces reliability. Each replica occupies 517 Kbytes or 1034 disk blocks of the slice.

The state database must be initialized before any metadevices are configured. See Chapter 2, “Installation and Setup” in the *Solstice DiskSuite 4.0 Administration Guide* for information about setting up the initial state database.

Performing the Memory Service

After you have configured metadevices, the metadvice driver must “remember” this configuration and status information. The metadvice state database is the metadvice driver’s long-term memory. The metadvice driver stores all the metadvice configuration information in the state database. This includes the configuration information about metadevices, mirrors, trans devices, RAID devices, and hot spares.

If the replicated metadvice state databases were lost, the metadvice driver would have no way of knowing any configuration information. This could result in the loss of all data stored on metadevices. To protect against losing the metadvice state database because of hardware failures, multiple replicas (copies) of the state database are kept.

These multiple replicas also protect the state database against corruption that can result from a system crash. Each replica of the state database contains a checksum. When the state database is updated, each replica is modified, one at a time. If a crash occurs while the database is being updated, only one of the

replicas will be corrupted. When the system reboots, the metadvice driver uses the checksum embedded in the replicas to determine if a replica has been corrupted. Corrupted replicas are ignored.

If a disk containing the metadvice state database is turned off, the metadvice remain fully functional because the database is retrieved from one of the replicas still in operation. Changes made to the configuration following the reboot are stored only in those replicas that are running when the system comes back up. If the disk drive that was turned off is later turned back on, the data contained in the replica stored on that disk will be ignored.

Planning Locations of Replicas

Consider the following when planning the location of replicas on a system where DiskSuite has just been loaded:

- Database replicas can reside on any unused partition or on any partition which will also be part of a metadvice or logging device with the exception of root (/), swap, /usr, or an existing file system.
- If multiple controllers exist, replicas should be spread as evenly as possible across the controllers.
- If multiple disks exist on a controller, at least two of the disks on each controller should store a replica of the metadvice state database.
- At least three replicas should be created. For instance, if you have three drives, one replica should be created on each. That way, if one drive fails, you will have the necessary two replicas to continue running. When you have less than two replicas, DiskSuite will not function.

Note – In a two-drive configuration, you should always create two replicas on each drive. For example, assume you create two replicas on one component and only one replica on the other. If the component with two replicas fails, DiskSuite will not function because the remaining component only has one replica.

- No more than one replica should be placed on a single disk unless that is the only way to reach the minimum number (three) of replicas. If you must have more than one replica on a slice, use the Metadvice State Database Information window to attach the replicas. See page 231 for additional information.

Creating the Initial State Database Replicas

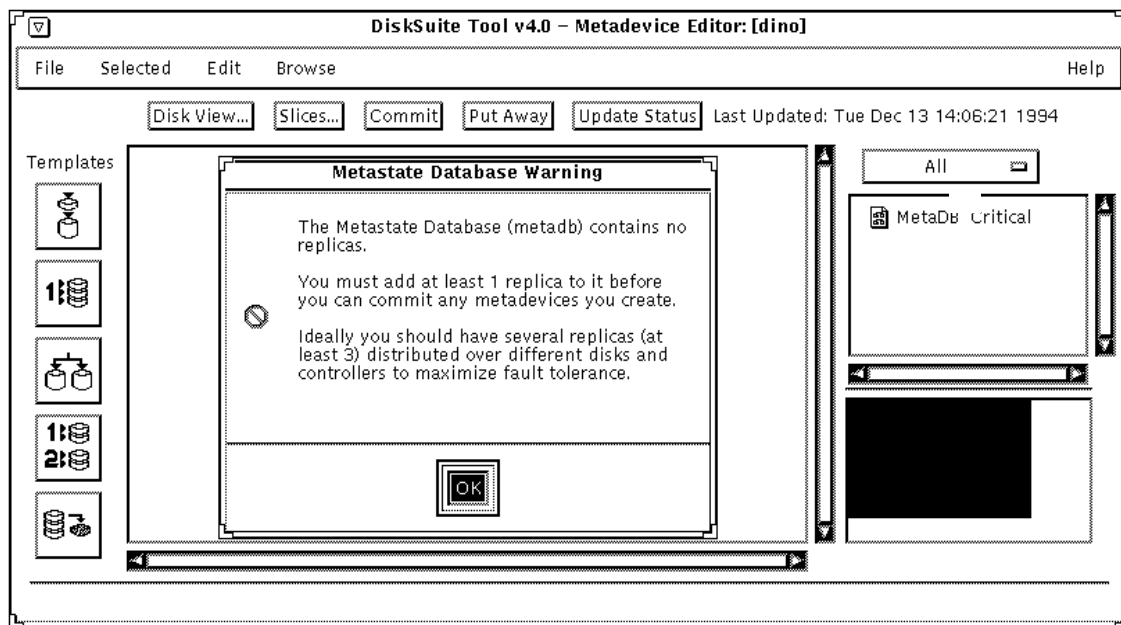
The following instructions explain the procedure for creating the initial state database and adding two replicas, creating the minimum of three.

1. Start DiskSuite Tool.

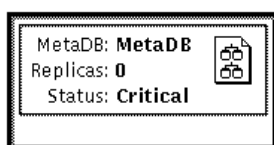
```
# metatool
```

2. Respond to the warning message.

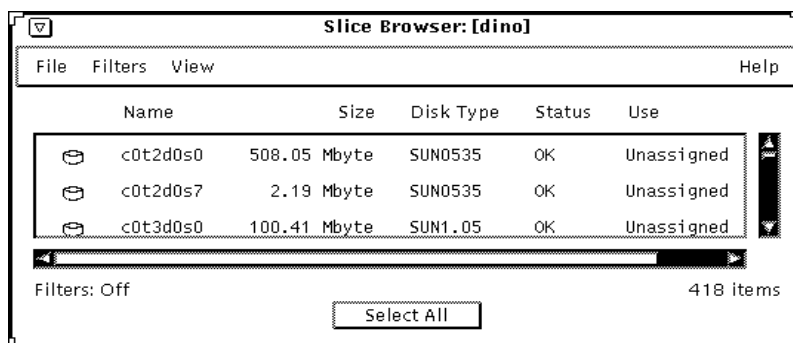
The first time DiskSuite Tool is started on a system, it displays a warning message. Click on the OK button.



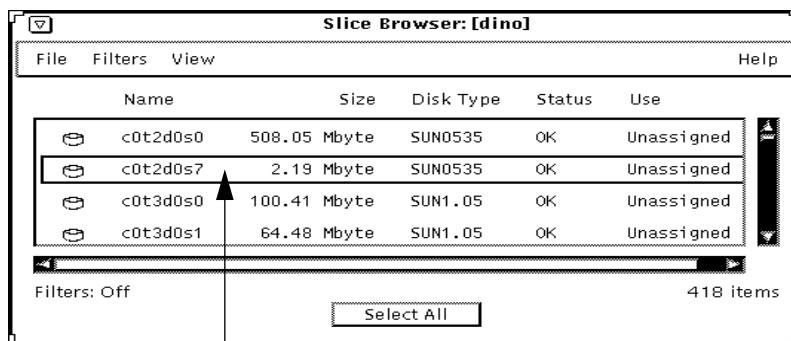
3. **Open the MetaDB template on the Metadevice Editor window's canvas.**
Double-click on the MetaDB object in the device list. The MetaDB object is displayed with a Critical status.



4. **Open the Slice Browser.**
Click on Slices on the button panel.



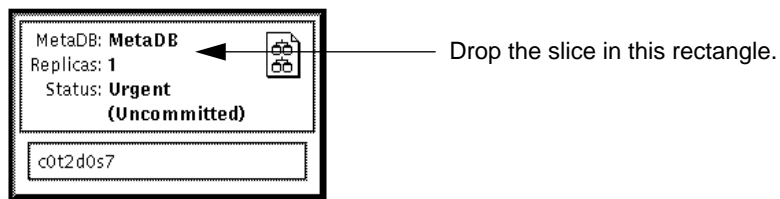
5. **Select a slice that will contain the state database.**
Scroll through the Slice Browser and find a slice that will contain the state database. Ideally, you should have a slice that is approximately 2 Mbytes. However, if all the slices are larger, you should select a slice that you are sure will be part of a metadevice. Unused space on the slice can only be used in a metadevice.



Click here to select the slice.

6. Drag the initial state database slice into the MetaDB template.

Point to the slice in the Slice Browser, press the middle button, and drag the slice to the MetaDB template. The status changes from Critical to Urgent.

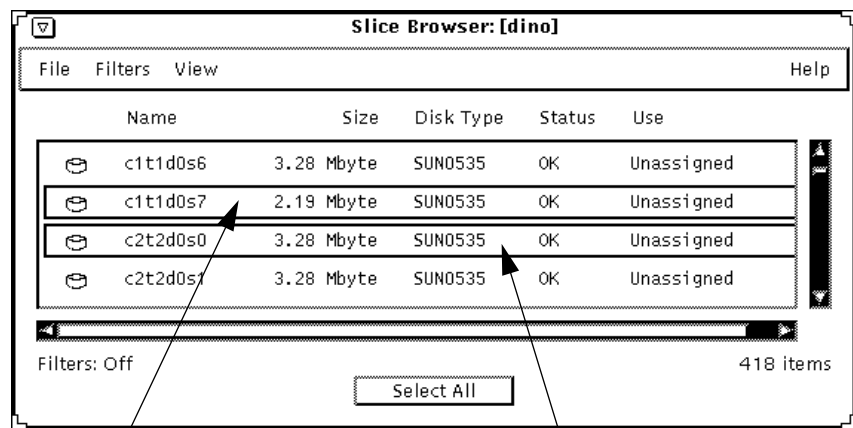


7. Select additional slices that will contain replicas of the state database.

Scroll through the list of devices in the Slice Browser and locate a minimum of two additional slices that will contain replicas of the state database. Each slice should be approximately 2 Mbytes.

Note – If the replicas and the state database are on slices on the same controller, the MetaDB will have a status of Attention. Also, you run the risk of lost data due to a possible single-point hardware failure.

After you have located appropriate slices to be replicas, click on the first slice, then hold down the Control key and click on the second slice.

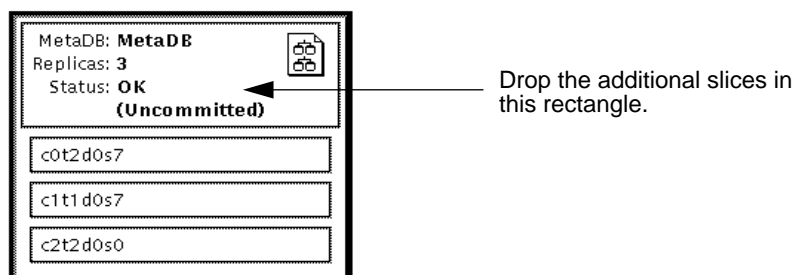


Click here to select the first slice.

Hold down the Control key and click here to select the second slice.

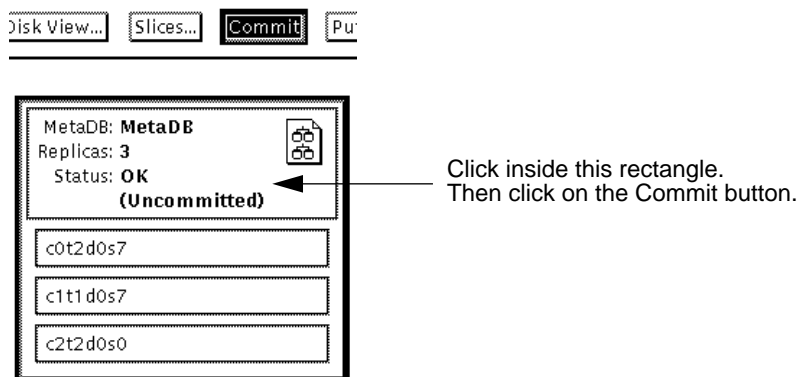
8. Drag the two slices into the MetaDB template.

Point to one of the two selected slices, press the middle button, and drag the slices to the top rectangle of the MetaDB object. The Status field changes from Urgent to OK.



9. Commit the MetaDB object.

Click on the top rectangle of the MetaDB object. Then click on Commit on the button panel.



Adding Replicas

Additional replicas, containing identical information, can be added to the MetaDB object at any time. The additional replicas ensure the integrity of the DiskSuite configuration.

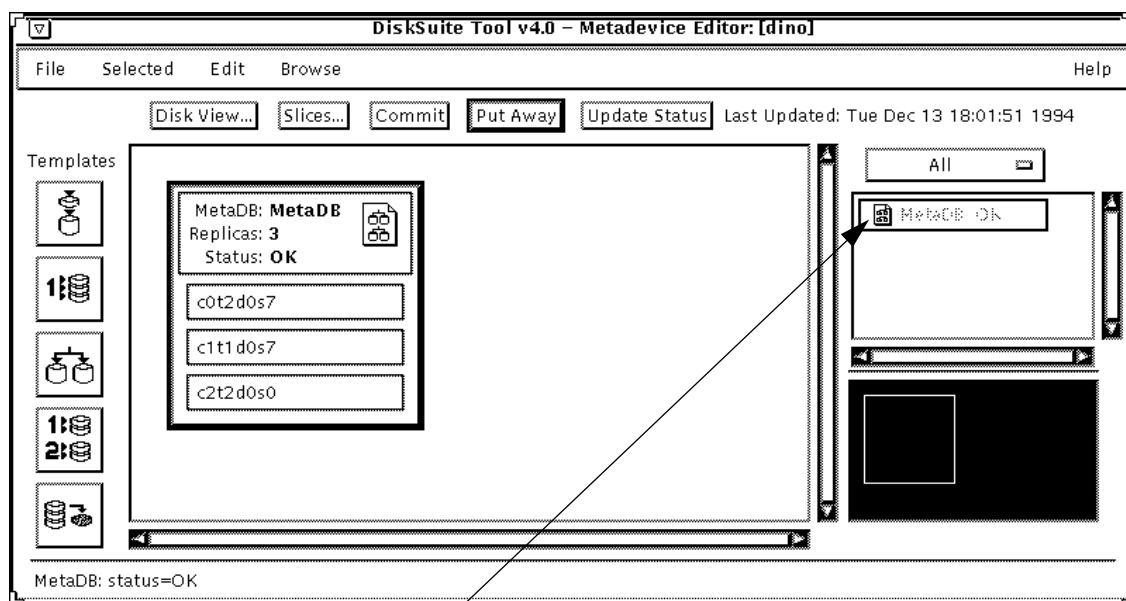
The following instructions explain the procedure for adding replicas.

1. Start DiskSuite Tool.

```
# metatool
```

2. Open the MetaDB object.

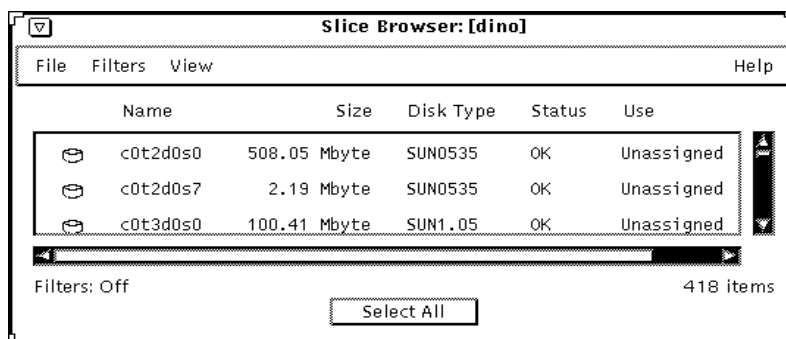
Point to the MetaDB object and double-click the left button.



Double-click here to open the MetaDB object.

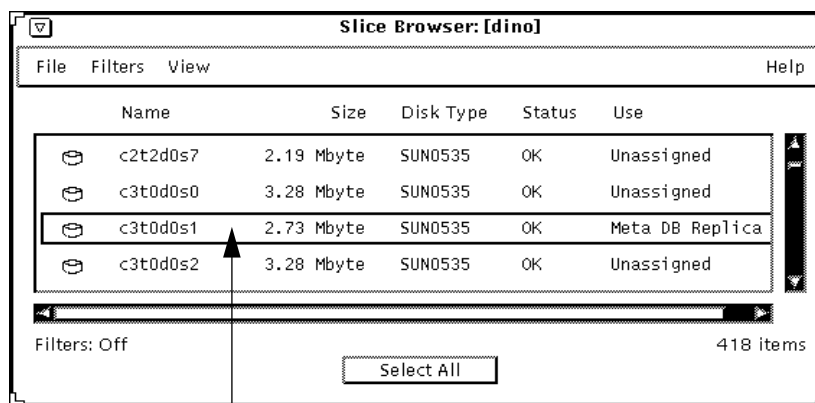
3. Open the Slice Browser.

Click on Slices on the button panel.



4. Select a slice for the new replica.

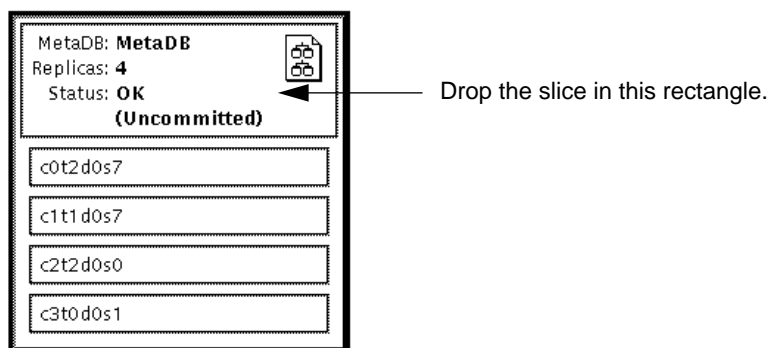
Scroll through the list of devices on the Slice Browser to locate a slice. The slice should be approximately 2 Mbytes.



Click here to select the slice.

5. Drag the slice into the MetaDB object.

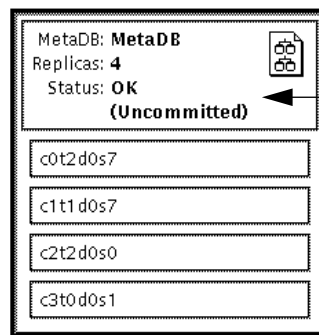
Point to the slice, press the middle button, and drag it to the top rectangle of the MetaDB object.



6. Commit the MetaDB object.

Click inside the top rectangle of the MetaDB object. Then click on the Commit button on the button panel.

Disk View... Slices... **Commit** Put



Click inside this rectangle.
Then click on the Commit button.

Removing Replicas

DiskSuite Tool lets you remove all but the last replica from the state database.



Caution – Don't remove all state database replicas if metadevices are still configured. If all replicas are removed, all metadevices will become inoperable.

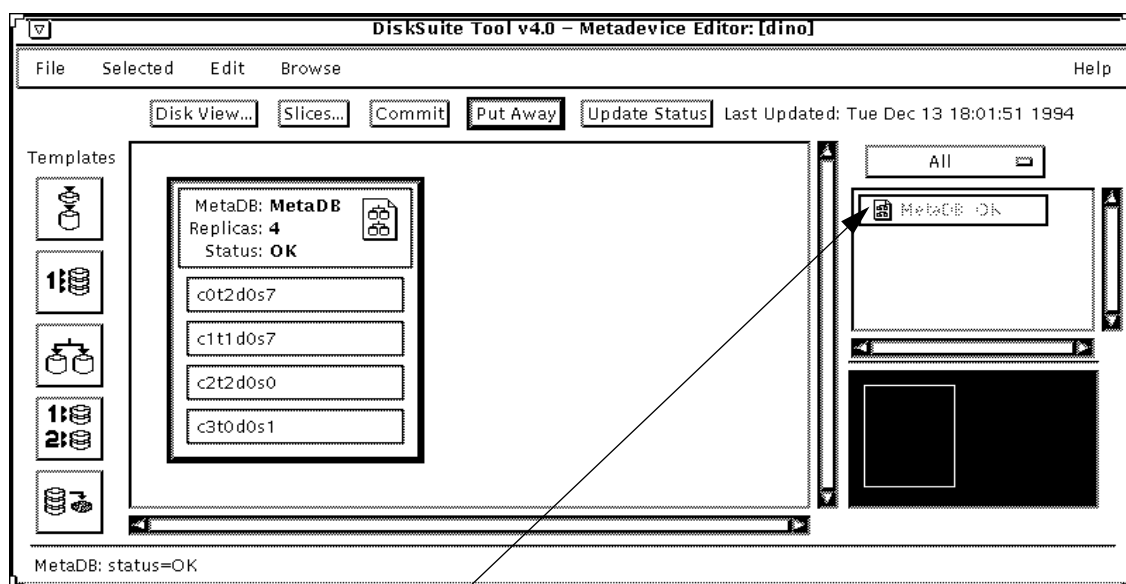
To remove a replica from the state database, use the following instructions.

1. Start DiskSuite Tool.

```
# metatool
```

2. Open the MetaDB object.

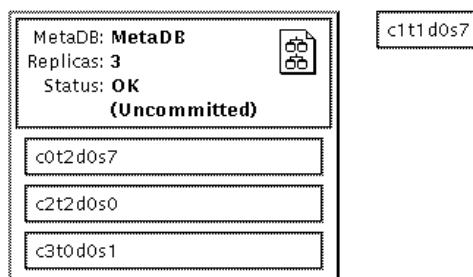
Point to the MetaDB object in the device list region of the Metadevice Editor window and double-click the left button.



Double-click here to open the MetaDB object.

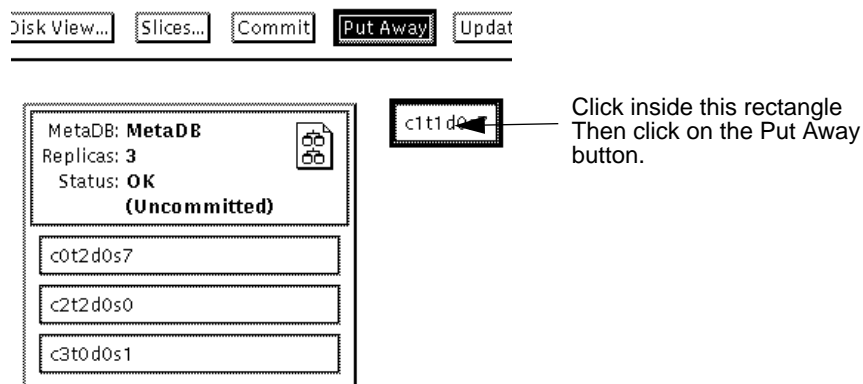
3. Drag the replica from the MetaDB object.

Point to the replica that you want to remove, press the middle button, and drag the slice to the Metadevice Editor's canvas.



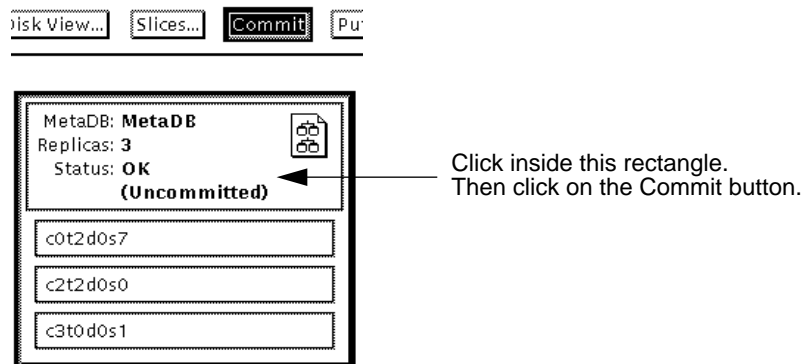
4. Put the slice away.

Click on the c1t1d0s7 slice. Then click on Put Away on the button panel. The slice is returned to the Slice Browser and has a use of unassigned.

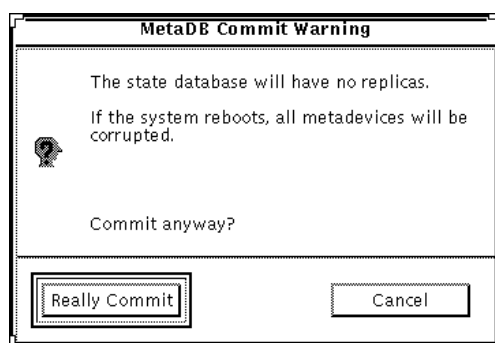


5. Commit the MetaDB object.

Click on the top rectangle of the MetaDB object. Then click on Commit on the button panel.



Note – The following dialog box is displayed if you attempt to remove the state database and all the replicas.



Metadevice State Database Information Window

The Metadevice State Database Information window enables you to view and modify the attributes and components of metadevice state database replicas. Figure 10-1 shows the Metadevice State Database Information window. There are three ways to display this window:

- Point to the MetaDB in the Metadevice Browser and double-click the left button. The MetaDB object is opened on the Metadevice Editor's canvas and the Metadevice State Database Information window is displayed.
- If the MetaDB object is on the Metadevice Editor's canvas, point inside the top of the rectangle and bring up the MetaDB menu. Select the Info choice.

- Double-click on a MetaDB object on the Metadevice Editor's canvas.

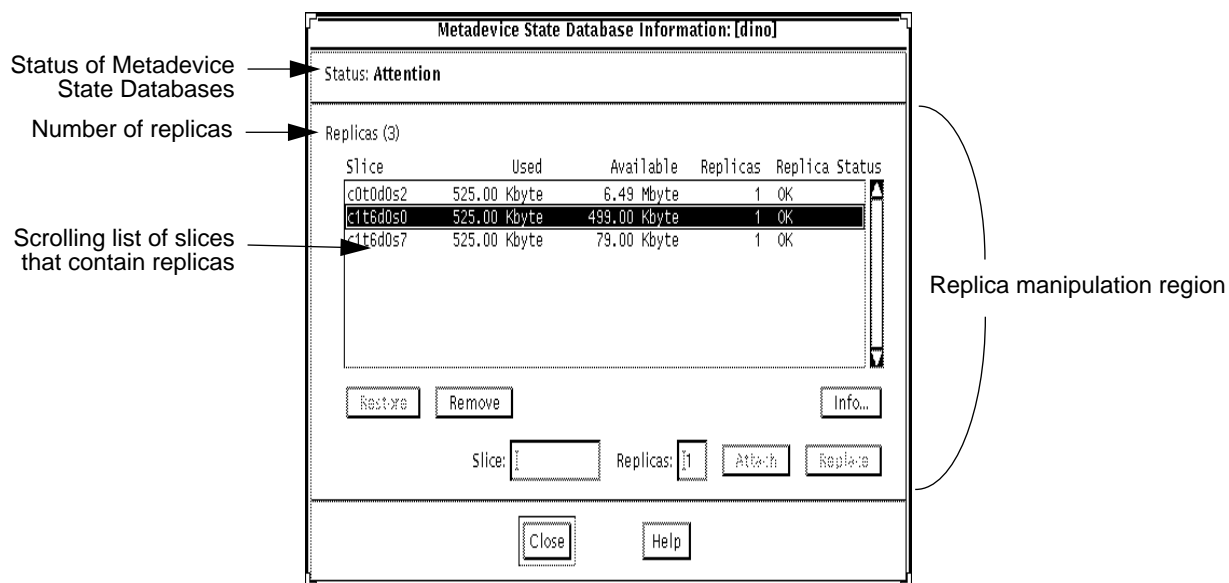


Figure 10-1 Metadevice State Database Information Window

Once changes are made, the MetaDB object must be committed before the changes take effect.

Table 10-1 lists the functionality associated with the regions of the Metadevice State Database Information window.

Table 10-1 Metadevice State Database Information Window Functionality

Field	Functions
Status	Description of the metadevice state database's status. See "Checking the Status of Replicas" on page 234" for a description of the values that appear in this region.
Replica manipulation region	<p>This region shows the following information and allows for manipulation.</p> <p>Replicas – Shows the number of replicas.</p> <p>Scrolling List – A scrolling list of the slices that contain replicas. The information in this field includes the name of the slice, the amount of space used, space available, number of replicas on the slice and the replica status.</p> <p>Remove – Removes the selected slices.</p> <p>Restore – Restores the selected slices if they are in error.</p> <p>Info – Displays the Slice Information window for the selected (highlighted) slice.</p> <p>Slice – This field is used to specify a new slice to attach to the MetaDB or replace the selected slice.</p> <p>Replicas – Shows the number of replicas that will be created on the slice. This value is set to one by default.</p> <p>Attach – Adds the slice entered in the slice field to the Replica list. This button is available only when a slice name is entered.</p> <p>Replace – Replaces the selected slice with the slice entered in the Slice field.</p>

Checking the Status of Replicas

There are three ways to check the status of the state database and the replicas using DiskSuite Tool. The methods are:

- Double-click on the MetaDB object in the device list and view the status on the Metadevice State Database Information window.
- Move the MetaDB object from the device list and view the Status field on the object.
- Bring up the Problem List, which is available using the Browse menu on the menu bar of the Metadevice Editor window. If a problem exists, a detailed description of the problem is given.

Note – The `metadb(1M)` command, which is documented in the *Solstice DiskSuite 4.0 Administration Guide*, displays additional information about the state database replicas.

The Status fields of the MetaDB object use the status keywords explained in Table 10-2.

Table 10-2 MetaDB Status Keywords

Keyword	Meaning
OK	The MetaDB has no errors and is functioning correctly.
Attention	There is only one replica present in the configuration. This status will also be displayed if you have the metadevice state database replicas on fewer than three different controllers.
Urgent	There is a state database present, but there are no replicas in the configuration.
Critical	There is no state database present in the configuration.

Expanding a File System

11 

This chapter provides information about file system expansion using Solstice DiskSuite Tool (`metatool(1M)`). Refer to Chapter 3, “Overview of DiskSuite Tool,” for information on using DiskSuite Tool.

Use the following table to locate specific information.

<i>File System Expansion Overview</i>	<i>page 235</i>
<i>Nonexpandable File Systems</i>	<i>page 236</i>
<i>Growing File Systems</i>	<i>page 236</i>

File System Expansion Overview

You can expand a mounted UFS by using DiskSuite Tool. Only UNIX file systems can be expanded using the DiskSuite Tool software.

Expanding a file system occurs in two steps.

1. Disk space is added at the end of the metadvice by using the DiskSuite concatenation facilities.
2. The file system is expanded using the DiskSuite Tool.

Note – Once a file system is expanded, it cannot be shrunk.

When you add space to mirrors, expand each of the submirrors before the file system. If an error is reported while expanding any of the submirrors, you must resolve that error before you can expand the file system.

For trans devices, only the logging device or master device can be expanded; not the trans device itself.

Nonexpandable File Systems

A mounted file system cannot be expanded if any of the following conditions exist:

- When `acct(1M)` is activated and the accounting file is on the target file system.
- There is a local `swap` file in the target file system.
- C2 security is activated and the logging file is on the target file system.
- The file system is `root (/)`, `/usr`, or `swap`.

Growing File Systems

When a component is available, DiskSuite Tool can be used to add the slice to the existing metadvice without interrupting service.

If more than one component is added, those components can be striped. This stripe is then concatenated onto the metadvice.

Disk space can be expanded by one or all of the following:

- Adding a slice (physical partition) to a metadvice.
- Adding a slice to a RAID metadvice.
- Adding a slice (physical partition) or stripe to a submirror.

To expand disk space, use the instructions in the following subsections.

Adding a Slice to a Metadvice

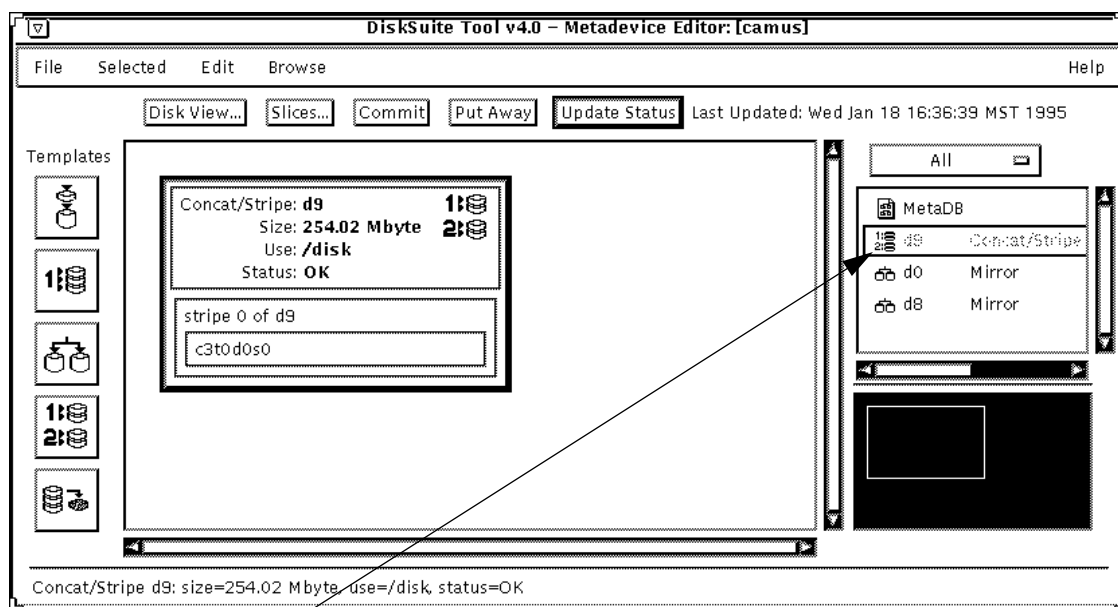
The following example provides step-by-step instructions for adding a slice to an existing metadvice. The metadvice will be a Concat/Stripe object that contains a mounted file system.

The steps for adding a slice to the existing metadvice follow:

1. Start DiskSuite Tool.

```
# metatool
```

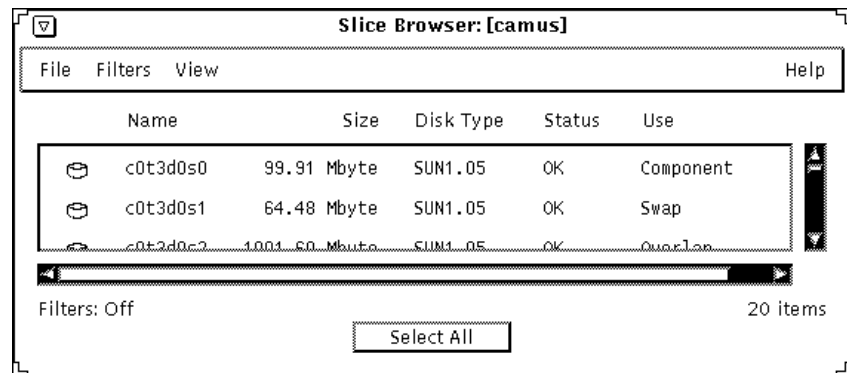
- 2. Open the Concat/Stripe object on the Metadvice Editor canvas.**
 Scroll through the device list on the Metadvice Editor to find the Concat/Stripe object that will receive an additional slice. In this example, the d9 Concat/Stripe object contains one slice, c3t0d0s0.



Double-click here to open the Concat/Stripe.

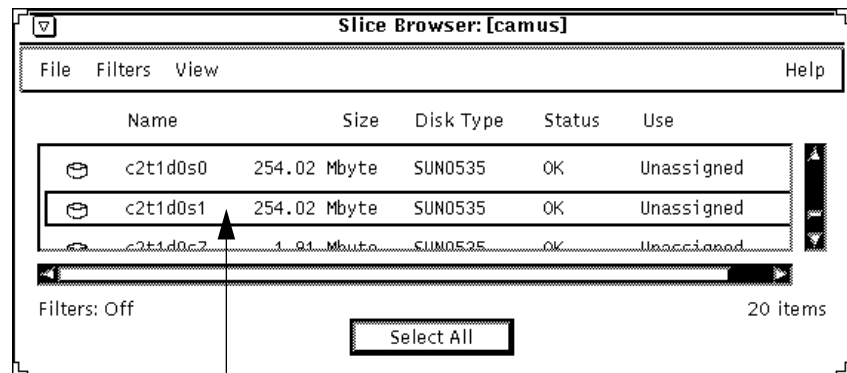
3. Open the Slice Browser.

Click on Slices on the button panel.



4. Select a slice to be added to the Concat/Stripe.

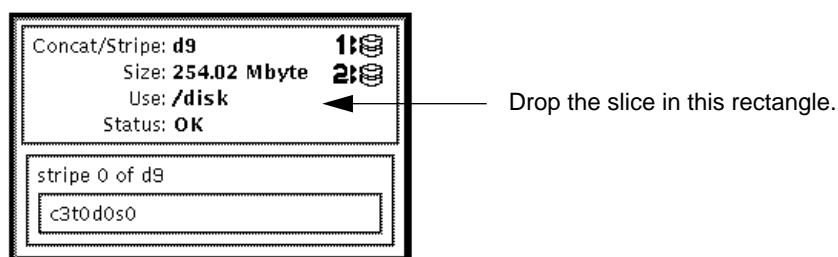
Select a slice of the appropriate size to add to the d9 metadvice. In this example, slice c2t1d0s1 is selected. (You can select additional slices at the same time that can be added to the Concat/Stripe.)



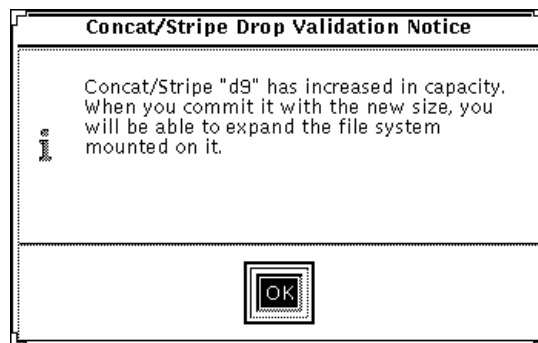
Click here to select the slice.

5. Drag the slice into the Concat/Stripe object.

Point to the `c2t1d0s1` slice, press down the middle button, and drag the slice to the top of the `d9` Concat/Stripe object.



The following validation dialog box is displayed when the slice is added to the Concat/Stripe. Click on the OK button.



6. Commit the Concat/Stripe object.

Click inside the top rectangle of the d9 Concat/Stripe object. Then click on Commit on the button panel.

Disk View... Slices... **Commit** Put Away

Concat/Stripe: d9		1:
Size: 508.05 Mbyte		2:
Use: /disk		
Status: OK		
(Uncommitted)		
stripe 0 of d9		
<input type="text" value="c3t0d0s0"/>		
stripe 1 of d9		
<input type="text" value="c2t1d0s1"/>		

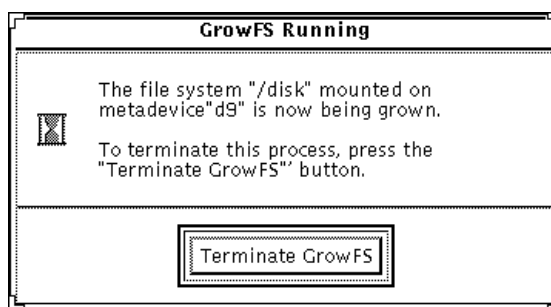
Click inside this rectangle.
Then click on the Commit button.

7. Expand the file system.

When the following dialog box is displayed, you can either expand the file system now or grow it manually later. In this example, Grow Now is selected.



The following dialog box lets you terminate the expansion. If you do not click on the Terminate GrowFS button, the expansion takes place.



8. The file system begins to use the added space.

The file system is using the additional space. For additional information about `growfs(1M)` and file system expansion, refer to Chapter 11 of the *Solstice DiskSuite 4.0 Administration Guide*.

Adding a Slice to a RAID Metadevice

The following example provides step-by-step instructions for adding a slice to an existing RAID device. The RAID device contains the `/disk` file system.

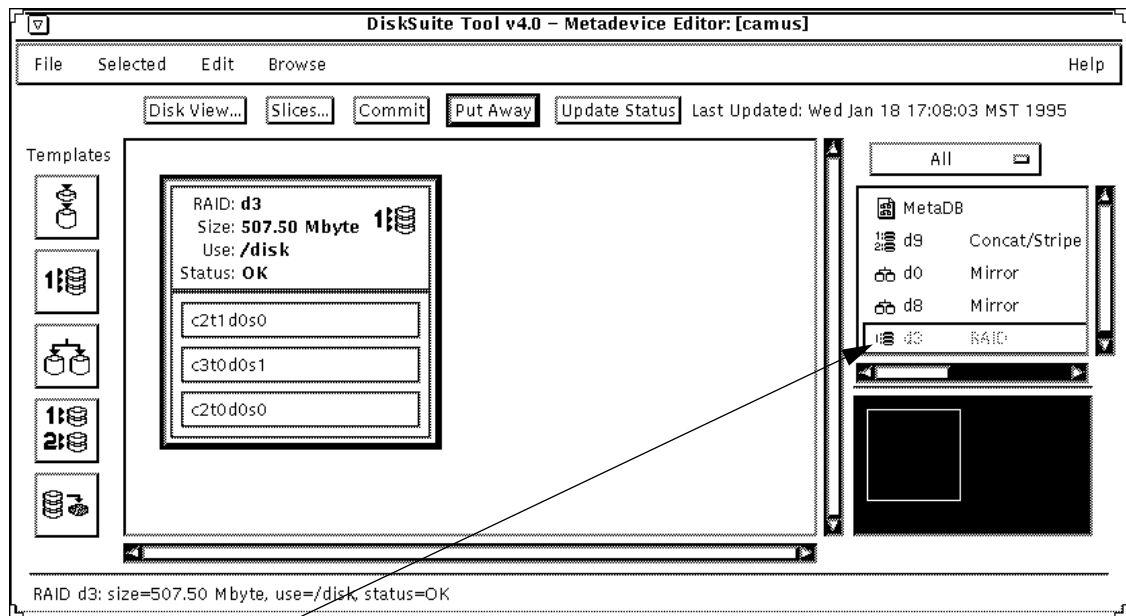
The steps for adding a slice to an existing metadevice follow:

1. Start DiskSuite Tool.

```
# metatool
```

2. Open the RAID device on the Metadevice Editor canvas.

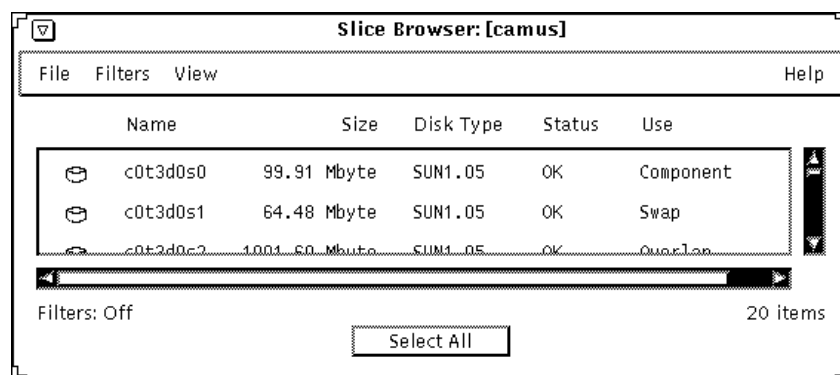
Scroll through the device list on the Metadevice Editor to find the RAID device that will receive an additional slice.



Double-click here to open the RAID.

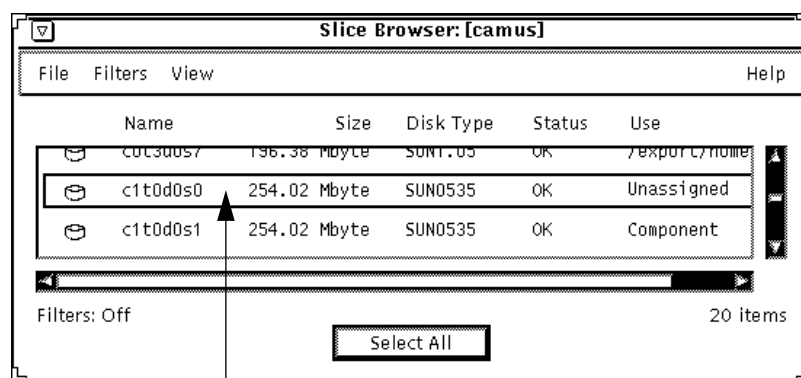
3. Open the Slice Browser.

Click on Slices on the button panel,



4. Select a slice to be added to the RAID device.

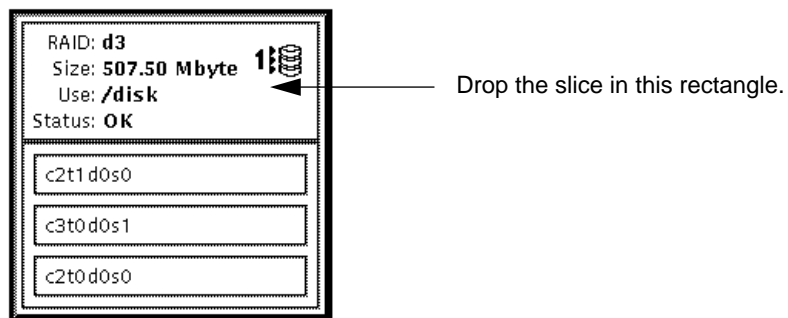
Select a slice of the appropriate size to add to the d3 RAID device. In this example, slice c1t0d0s0 is selected. (You can select additional slices at the same time that can be added to the RAID device.)



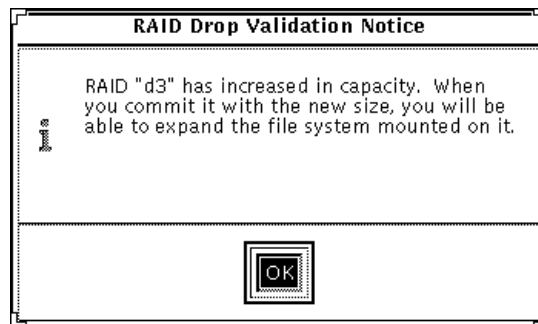
Click here to select the slice.

5. Drag the slice into the RAID device.

Point to the `c1t0d0s0` slice, press the middle button, and drag the slice to the top rectangle of the `d3` RAID device.



The following validation dialog box is displayed when the slice is added to the RAID device. Click on the OK button.




When you click on the OK button, the slice is added to the RAID device.

6. Commit the RAID device.

Click on the top rectangle of the d3 RAID device. Then click on Commit on the button panel.

Disk View... Slices... **Commit** Pl

RAID: d3 Size: 761.58 Mbyte Use: /disk Status: OK (Uncommitted)	
c2t1 d0s0	
c3t0 d0s1	
c2t0 d0s0	
c1t0 d0s0	

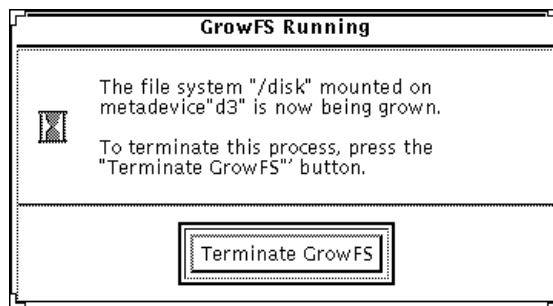
Click inside this rectangle.
Then click on the Commit button.

7. Grow the file system.

When the following dialog box is displayed, you can either expand the file system now or grow it manually later. In this example, Grow Now is selected.



The following dialog box lets you terminate the expansion. If you do not click on the Terminate GrowFS button, the expansion takes place.



8. The RAID device begins to use the added space.

The RAID device is using the additional space. For more information about `growfs` and file system expansion, refer to Chapter 11 of the *Solstice DiskSuite 4.0 Administration Guide*.

Adding Slices to Submirrors

The following example provides step-by-step instructions for adding a slice to an existing mirror. In this example, the `d0` mirror contains two submirrors, `d1` and `d2`.

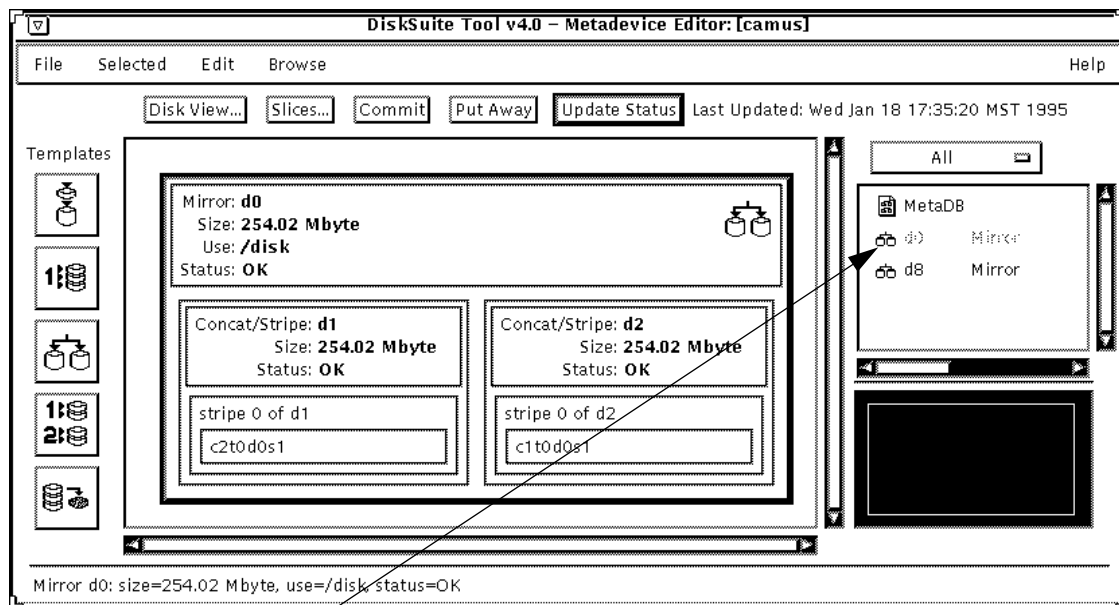
The steps for adding a slice to an existing mirror follow:

1. Start DiskSuite Tool.

```
# metatool
```

2. Open the `d0` mirror on the Metadevice Editor canvas.

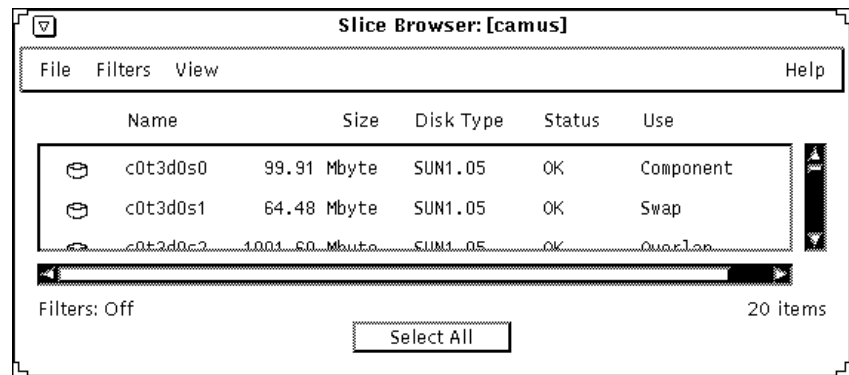
Scroll through the device list on the Metadevice Editor to find the `d0` mirror object that will receive additional slices.



Double-click here to open the `d0` Mirror.

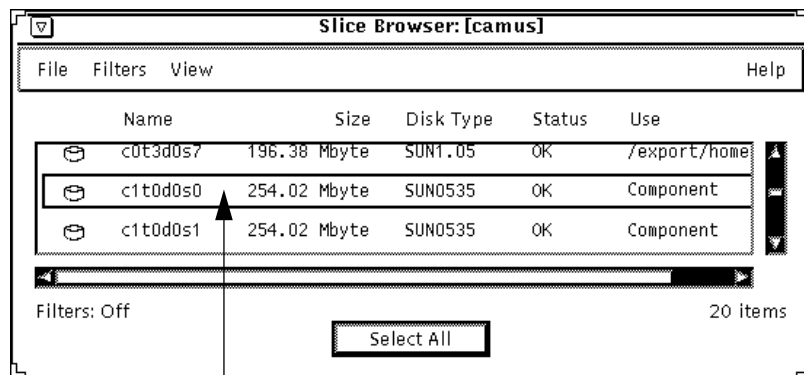
3. Open the Slice Browser.

Click on Slices on the button panel.



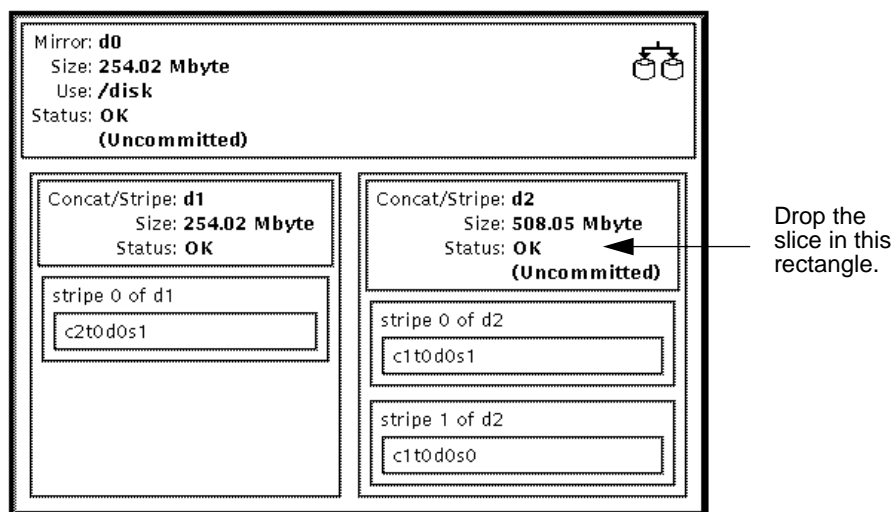
4. Select a slice to be added to the d2 Concat/Stripe (submirror).

Select a slice of the appropriate size to add to the d2 Concat/Stripe (submirror). In this example, slice c1t0d0s0 was selected. (You can select additional slices at the same time that can be added to the d2 Concat/Stripe.)

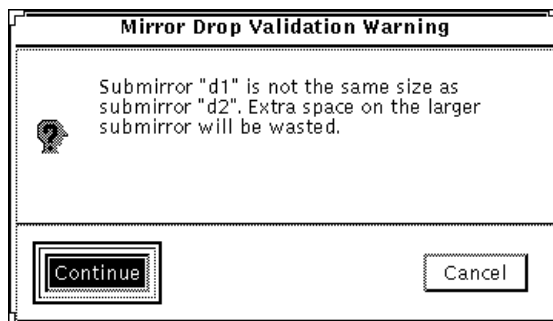


5. Drag the slice into the d2 submirror.

Point to the selected slice (c1t0d0s0), press the middle button, and drag the slice to the top of the d2 submirror. The slice is concatenated to the bottom of the Concat/Stripe.

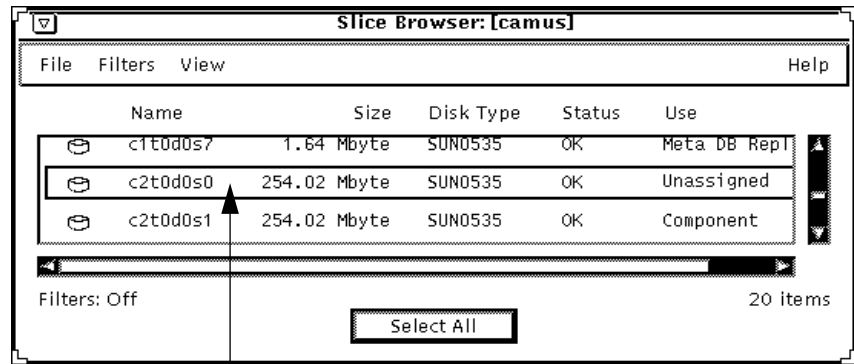


When the slice is dropped, the following warning dialog box is displayed because the submirror will be a different size after adding the slice. Click on the Continue button.



6. Select a slice to be added to submirror d1.

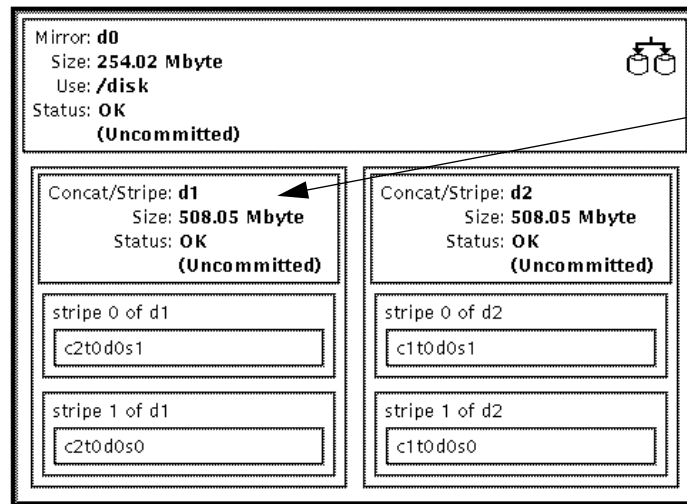
Scroll through the list of slices in the Slice Browser and locate a slice that is equal in size to the slice you just added to the d2 submirror. In this example, slice c2t0d0s0 is selected.



Click here to select the slice.

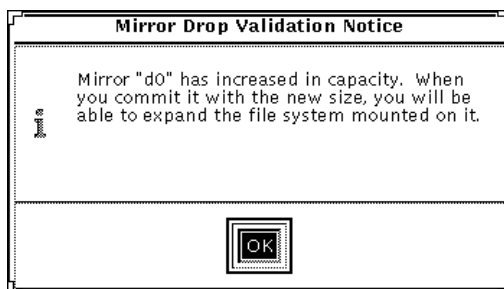
7. Drag the slice into the d1 submirror.

Point to the selected slice (c2t0d0s0), press the middle button, and drag the cursor to the top of the d1 submirror. The slice is concatenated to the bottom of the Concat/Stripe.



Drop the slice in this rectangle.

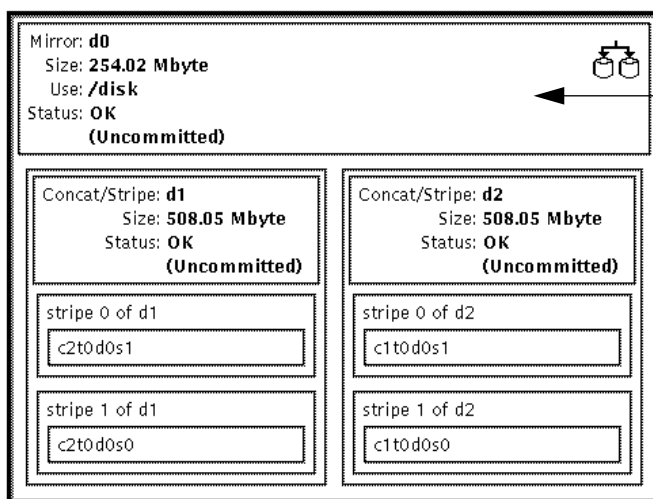
When the slice is dropped, the following Validation dialog box is displayed. Click on the OK button.



8. Commit the mirror.

Click on the top rectangle of the d0 Mirror. Then click on Commit on the button panel.

Last Updated: w



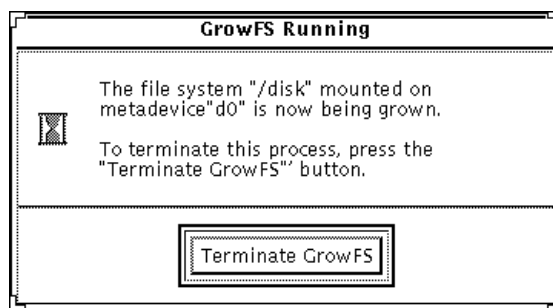
Click inside this rectangle. Then click on the Commit button.

9. Grow the file system.

When the following dialog box is displayed, you can either expand the file system now or grow it manually later. In this example, Grow Now is selected.



The following dialog box lets you terminate the expansion. If you do not click on the Terminate GrowFS button, the expansion takes place.



10. The file system begins to use the added space.

The Mirror is using the additional space. For more information about growfs and file system expansion, refer to Chapter 11 of the *Solstice DiskSuite 4.0 Administration Guide*.

Example Use of DiskSuite Tool



This appendix provides an example configuration and explains how to set up the example using DiskSuite Tool (`metatool(1M)`). Refer to Chapter 3, “Overview of DiskSuite Tool,” for information on using DiskSuite Tool.

Use the following table to locate specific information in this appendix.

<i>Overview of the Example</i>	<i>page 253</i>
<i>Planning for Availability</i>	<i>page 256</i>
<i>Setting Up the Configuration Using DiskSuite Tool</i>	<i>page 262</i>

Overview of the Example

This example shows how to take advantage of DiskSuite features that provide higher availability, greater capacity, and improved performance.

- **Higher availability** – In the example, mirroring ensures data will remain available even if a system board, controller, or disk drive fails. UFS logging significantly shortens the time the data is unavailable after scheduled downtime, a system crash, or a power failure.
- **Greater capacity** – Concatenation of disks, RAID, and striping increase capacity and ease the administrative load by combining many disks into a few large metadevices.

- **Improved performance** – UFS logging (trans devices) helps performance by decreasing the number of synchronous disk writes. Striping aids performance by automatically spreading the disk traffic across multiple disks.

The example uses the following:

- Software
 - The Solaris 2.x distribution
 - Third-party packages
 - DiskSuite and DiskSuite Tool packages
- File Systems
 - One Gbyte of system data and third-party packages
 - Four Gbytes of user data
- Databases
 - Four Gbytes of read-write data
 - Six Gbytes of read-mostly data
- Hardware (see Figure A-1)
 - One SPARCserver 1000™ (with two internal 535-Mbyte disks and four system boards)
 - Three Disk Tower 1000s (each with 16 535-Mbyte disks)

The example uses the following functionality from DiskSuite:

- Metadevice state database replicas
- Concatenations
- Stripes
- Mirrors
- RAID
- UFS logging
- Hot spares

Figure A-1 illustrates the hardware configuration.

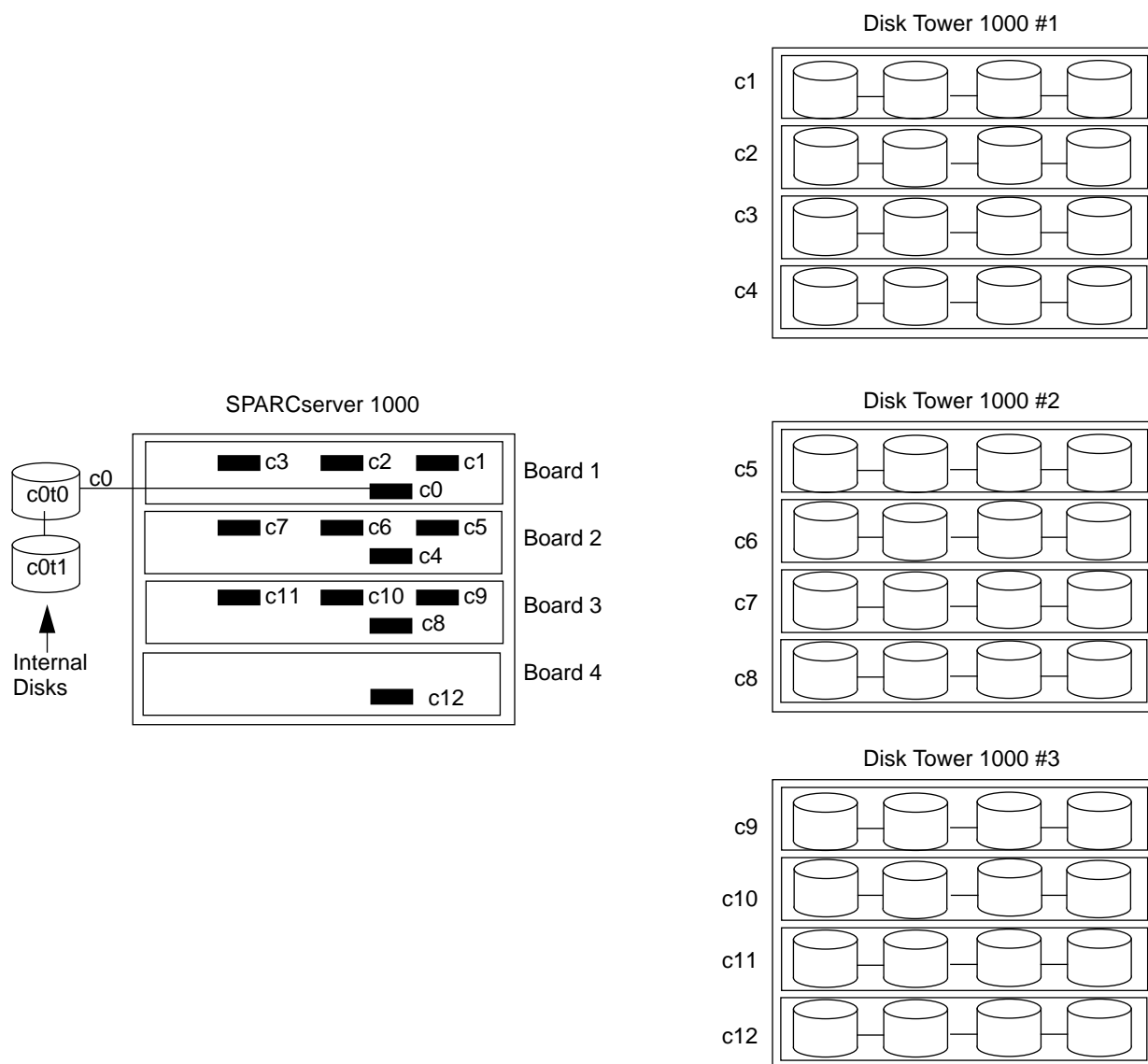


Figure A-1 Hardware View of the Example Configuration

Planning for Availability

The key to the DiskSuite type of availability is the replication or mirroring of data so that no one piece of hardware stands between the application's referencing the data and the media that contains the data.

When planning the configuration, availability of the data is achieved by:

- **Disk drive availability** – achieved by setting up mirrors across different disk drives.
- **Controller availability** – achieved by setting up mirrors across different controllers.
- **System board availability** – achieved by setting up mirrors across controllers on different system boards.

The hierarchy representation of the hardware shown in Figure A-2 helps illustrate the best availability when planning mirrors. An example use of the illustration would be to find a disk to mirror the data on `c4t0`. Selecting any other disk on any controller will ensure the data is available in the event of a single disk failure. But if you want the data available in the event of a controller failure, select a disk not connected to controller `c4`. Figure A-2 shows that any disk connected to the other controllers would keep the data available in the event of a disk or controller failure. Mirroring across system boards would further maximize data availability in this configuration. Because

c4 is connected to the second system board, selecting a disk attached to a different system board keeps the data available in the event of a system board, controller, or disk failure.

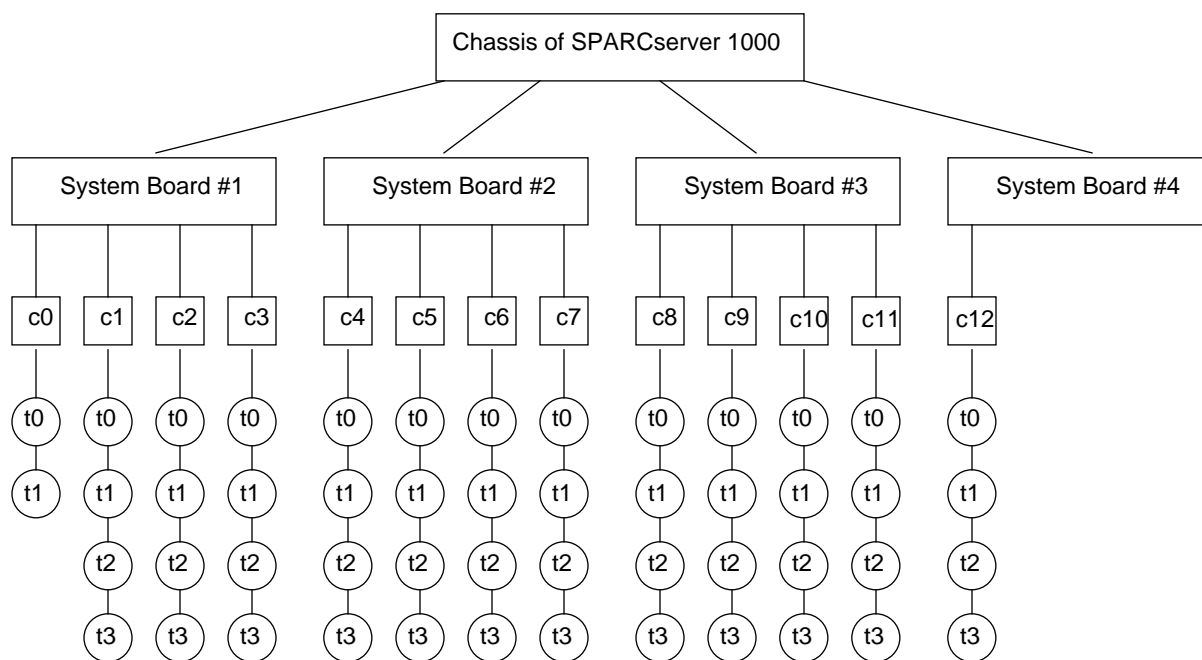


Figure A-2 Hierarchy of the Example Configuration

Planning the Layout of the Data

The following steps are used to plan the layout of the configuration:

1. Plan the location of Solaris 2.x and the third-party software.

The two internal 535-Mbyte disks, c0t0 and c0t1, will be used for the Solaris 2.x distribution, third-party software, and DiskSuite packages.

To provide greater availability, the two internal disks will be mirrored. The mirrors should be to disks on a different controller and, if possible, a different system board. Disks c12t0 and c12t1 were selected.

To further help availability, a 16-Mbyte slice will be reserved on one of the local disks for a UFS log. The `/usr` and `/opt` file system will share this log. Like the disks, the log will be mirrored. The log will be 16 Mbytes rather than the recommended minimum of one Mbyte because larger logs help performance on heavily used file systems. But the larger log will take a little longer to scan during reboot, perhaps as long as eight seconds. The log will be mirrored to help ensure availability.

2. Plan the location of the other file systems and databases.

The 535-Mbyte disks in the Disk Tower 1000s are too small to hold any one of the databases or file systems. Concatenation, striping and RAID devices will be used to create metadevices that are large enough to hold the data.

The file system that contains four Gbytes of user data will be a concatenation. To increase performance, the concatenation will be spread across as many controllers as possible. All `t0` and `t1` disks on controllers `c4`, `c5`, `c6`, and `c7` will be used. Additionally, the data will be mirrored to disks attached to different controllers. In this case, all `t0` and `t1` disks on `c8`, `c9`, `c10`, and `c11` will be used as mirrors.

To increase availability by speeding system reboots, the file system will be logged. A 64-Mbyte slice will be allocated on `c8t0` for the log, rather than the recommended maximum of eight Mbytes because larger logs help performance on heavily used file systems. However, the larger log may slow reboots by as much as 24 seconds, while the log is scanned. The log will be mirrored to help ensure availability.

The 4-Gbyte read-write database will be striped to enhance performance. The data will be located on all `t2` and `t3` disks on controllers `c4`, `c5`, `c6`, and `c7`. To increase availability, the data will be mirrored to all `t2` and `t3` disks on controllers `c8`, `c9`, `c10`, and `c11`.

In this example, there is not enough disk space to mirror the 6-Gbyte read-mostly database. A mirror is preferable because it provides greater availability and has better write performance than a RAID device. However, due to space and cost constraints, creating a large RAID device is a viable alternative. A RAID device provides better availability than a simple concatenation or stripe. All the disks on controllers `c1`, `c2`, and `c3` will be used for the 6-Gbyte, read-mostly database. (It's important to keep in mind that the actual usable size of the RAID device will be only 5 1/2 Gbytes because the equivalent of one disk's worth of space is used for parity.)

3. Plan the location of hot spares.

The use of hot spares makes all the mirrors and the RAID device more available. Disks `c12t2` and `c12t3` will be used as hot spares.

4. Plan the location of metadatabase state database replicas.

A minimum of three metadatabase state database replicas are required in Solstice DiskSuite configurations. This ensures the necessary two replicas will be available in the event of a single replica failure. Because a failure could occur because of a disk, controller, or system board failure, replicas will be placed on disks attached to three different system boards to help ensure availability.

The state database replicas will share the same slice as the UFS logs. Because the logs are spread across different system boards, the availability goals are met. The replicas will be placed on `c0t0d0s7`, `c8t0d0s7`, and `c12t0d0s7`.

Figure A-3 shows the layout of the data in the example configuration.

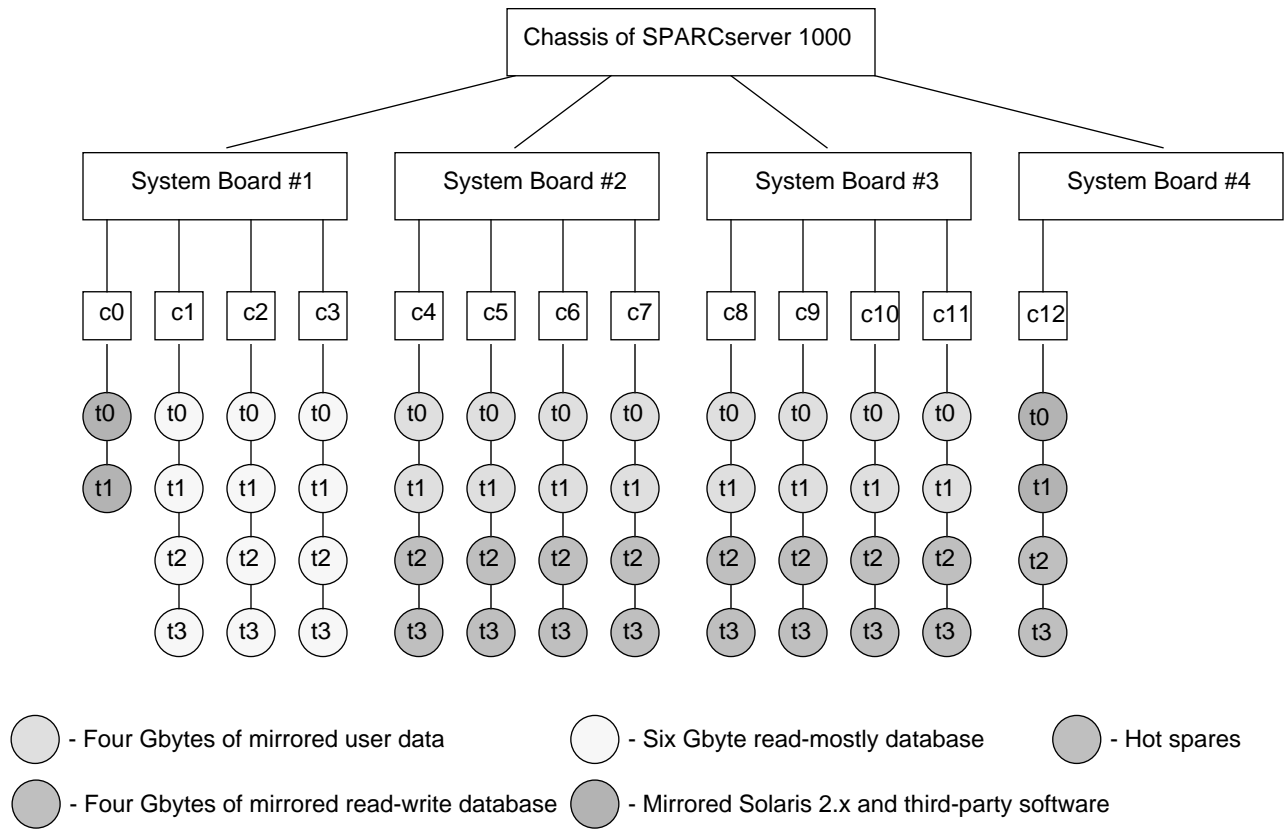


Figure A-3 View of the Data Layout

Figure A-4 shows a hardware view of the data layout.

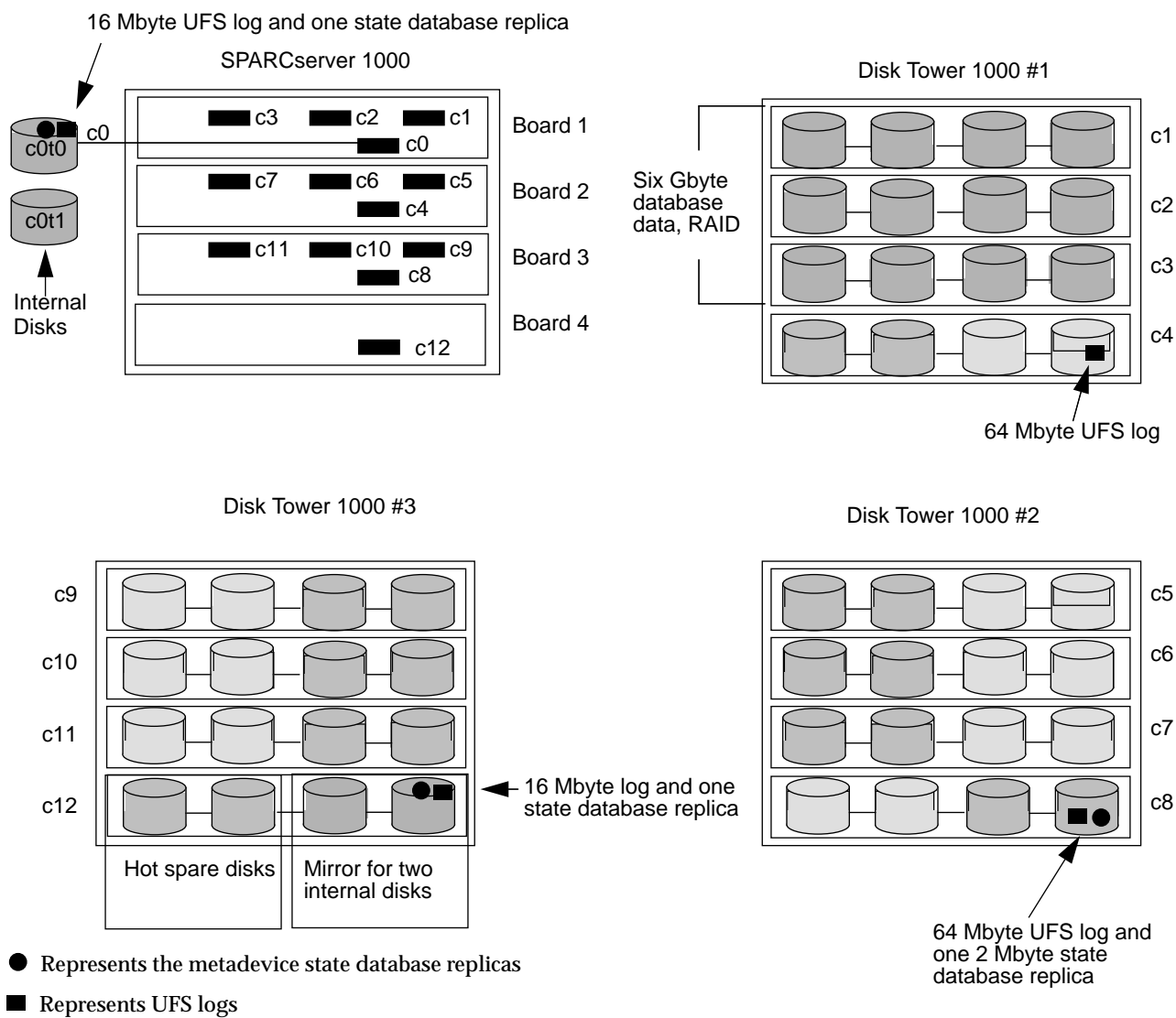


Figure A-4 Hardware View of the Layout

Setting Up the Configuration Using DiskSuite Tool

To use DiskSuite Tool to set up the configuration shown in Figure A-4, perform the following steps:

1. Partition the slices.

You should partition all slices before you start DiskSuite Tool. Use the `format(1M)` command to partition the disks.

Table A-1 Example Disk Partitioning

Slice	Size (Mbytes)	Use
c0t0d0s0	14.49	root (/)
c0t0d0s1	32.27	swap
c0t0d0s3	126.33	/usr/openwin
c0t0d0s4	87.50	/var
c0t0d0s5	131.66	/opt
c0t0d0s6	99.80	/usr
c0t0d0s7	17.64	State database (2 Mbytes), UFS log (16 Mbytes)
c0t1d0s2	--	/usr/opt, contains third-party software
c1, c2, and c3	--	RAID metadvicees, the entire disk is s2
c4t0d0s0	471	User data, mirror metadvicees
c4t0d0s7	64	UFS log
c4t1, c4t2, c4t3, c5, c6, and c7	--	User data, mirror metadvicees, the entire disk is formatted as s2.
c8t0d0s7	64	Mirror of UFS log
c8t0d0s6	2	State database
c8t1, c8t2, c8t3, c9, c10, and c11	--	User data, mirror metadvicees, the entire disk is formatted as s2.
c12t1d0s2	535	Mirror for c0t1d0
c12t2 and c12t3	--	Hot Spares
c12t0	--	Slice so it can be used to mirror data on c0t0.

2. Start DiskSuite Tool.

When the disks are partitioned you can start the DiskSuite Tool (`metatool(1M)`).

3. Create state database replicas.

When DiskSuite Tool starts, a warning message dialog box will display. The message is telling you there are no metadevice state database replicas. Point to the OK button and click the left button.

Use the instructions in the section “Creating the Initial State Database Replicas” on page 222 to create the state database replicas. The slices `c0t0d0s7`, `c8t0d0s7`, and `c12t3d0s7` will be used for the state database replicas.

4. Create mirrors.

You will first use the instructions in “Mirroring File Systems You Cannot Unmount” on page 110, to mirror the root (/), `/usr`, `/opt`, and `swap` file systems.

The remaining mirrors can be built using the instructions in “Creating Mirrors” on page 95. The four-Gbyte read-write database will be striped. You would use the instructions in “Striped Metadevices” on page 68 to see how to stripe. The four-Gbyte mirror of user data is a concatenation. You would use the instructions in “Concatenated Metadevices” on page 63 to learn how to build concatenations.

5. Create RAID device.

The six-Gbyte RAID device will be created using the instructions found in “Creating RAID Metadevices” on page 199.

6. Create UFS logging (Trans) devices.

You are logging two separate file systems. The 16-Mbyte log is for the `/usr` and `/opt` file systems and a 64-Mbyte log is for the four Gbytes of user data. To log the `/usr` and `/opt` file systems and the four-Gbyte user data, you would use the instructions in “How to Set Up UFS Logging on an Existing File System” on page 146.

7. Create hot spare pools.

To create the two hot spares using the `c12t2d0` and `c12t3d0` disks, use the instructions found in the section “Defining Hot Spares” on page 168.

8. Associate hot spare pools.

To associate the two hot spares with all the mirrors and the RAID use the instructions found in the section “Associating Hot Spare Pools” on page 172. You must repeat the instructions for each of the mirrors and RAID devices in the configuration.

9. Setup databases on the RAID device and the mirror.

Refer to the documentation that comes with the database product you are using. The RAID metadvice and mirror can be treated as any other device.

10. Run `newfs(1M)` on the file system that contains the user data.

When all the metadvice and hot spares are created, you must invoke the `newfs` command on the file system that contains the user data. For example:

```
# newfs /dev/md/rdisk/dn
```

In the above example, `dn` is the metadvice name of the Trans device where the four Gbytes of user data will reside.

11. Edit `/etc/vfstab` file.

Entries for the four Gbytes of mirrored user data must be made in the `/etc/vfstab` file. This entry will have the following format:

```
/dev/md/dsk/dn /dev/md/rdisk/dn /usr_data ufs 4 yes -
```

12. Export file systems.

Any file systems or databases that will be shared across the network must be exported. For instance if you want to export `/usr`, you would put a line similar to the following into the `/etc/dfs/dfstab` file:

```
share -F nfs -d "/usr" /dev/md/dsk/dn
```

This line may be different, depending on how your system is networked.

You would then export the `/usr` file system with the `shareall(1M)` command:

```
# shareall
```

DiskSuite Tool Messages



This appendix contains the status, error, and log messages displayed by DiskSuite and the DiskSuite Tool. Use the following table to locate specific information in this appendix.

<i>State Information Terms</i>	<i>page 265</i>
<i>Metadevice Editor Messages</i>	<i>page 266</i>
<i>Dialog Box Error Messages</i>	<i>page 267</i>
<i>Dialog Box Warning Messages</i>	<i>page 271</i>
<i>Dialog Box Information Messages</i>	<i>page 279</i>
<i>Metadevice Editor Window Messages</i>	<i>page 280</i>
<i>Disk View Window Messages</i>	<i>page 286</i>
<i>Log Messages</i>	<i>page 288</i>

State Information Terms

The dialog boxes shown in this appendix use the following terms when the state of the components is reported.

- **OK** – The component is operating properly.
- **Resyncing** – The component is in the process of resyncing (copying) the data.

- **Erred** – The component has encountered an I/O error or an open error. All reads and writes to and from this component have been discontinued. See “Replacing and Enabling Submirror Components” on page 134 for information on component replacement.
- **Last erred** – The component has encountered an I/O error or an open error. However, the data is not replicated elsewhere due to another component failure. I/O is still performed on the component. If I/O errors result, the mirror I/O will fail. See “Replacing and Enabling Submirror Components” on page 134 for information on component replacement.

Metadevice Editor Messages

The Metadevice Editor has a message line at the bottom of the window that displays information about objects and actions. There are two types of message line feedback displayed:

- When you are pointing the cursor at an object, the message line has the following format:

```
object_type object_name: attribute=value, . . .
```

- When you are dragging an object that is not yet populated, the message line has the form:

```
Drop requirement comp_type into new_object_type new_object_name
```

Once the object is sufficiently populated, the message line has the form:

```
Drop comp_type into new_object_type new_object_name or commit
```

Dialog Box Error Messages

DiskSuite Tool displays the following messages in Error dialog boxes. When these messages are displayed, the only choice on the dialog box is OK. The action you attempted will not be performed. Use the information here to correct the error.

```
A RAID can only have one operation that causes a resync per commit
```

You have tried to commit two separate changes to a RAID device at the same time. While the changes may be valid, only one can be performed at once. For example, if you replace a slice and add a new slice, this message is displayed. You must perform one change and click on the Commit button, then perform the other change and click on the Commit button.

```
Concat dn has no stripes
```

You have tried to commit a concatenation that has no stripes. You must add stripes to the concatenation.

```
Cannot delete a metadvice that is in use.
```

You have tried to delete a metadvice that contains a mounted file system.

```
dn has no components.
```

You have tried to commit a Concat/Stripe template that has no slices. You must add slices to the object before clicking on the Commit button.

Mirror *dn* has no submirrors

You have tried to commit a Mirror that has no submirrors. You must add submirrors before clicking on Commit.

RAID *dn* must have at least three parity slices.

You have tried to commit a RAID metadvice that has fewer than three slices. Add the necessary slices and commit the RAID metadvice.

Slices added to a RAID device must be at least as large as the smallest original slice.

You have tried to add a slice to a RAID device that is smaller than the slices that are already part of the RAID device.

Slice *slice* is mounted. You cannot add it to Concat/Stripe *dn*, it is not the first mounted slice to be added.

You have tried to add a slice that has a mounted file system to a Concat/Stripe and there is already at least one slice in the Concat/Stripe. The slice with the mounted file system must be the first one added to the Concat/Stripe.

Slice *slice* is mounted. You cannot add it to Concat/Stripe *dn*, the Concat/Stripe device has already been created.

You have either tried to add a slice that contains a mounted file system to a Concat/Stripe or RAID template. In the case of a Concat/Stripe, the slice that contains the mounted file system must be the first slice added. In the case of a RAID, you must first unmount the file system because the file system will be overwritten during a resync.

Slice *slice* is too small to use in a RAID device.

You have tried to add a slice that is too small. The slice being added is either smaller than the slices already in the RAID device or is too small to be used in a RAID device.

Submirror *dn* has a mounted file system, it should be the first submirror added. Mirror *dn* has a component with a file system mounted. You cannot add another submirror.

You have tried to add a submirror that contains a mounted file system and the Mirror already has a mounted file system on the other submirror. You must add an unassigned slice.

Submirror *dn* is too small.

You have tried to add a new submirror that is smaller than the current size of the Mirror. The submirror must be as large as the existing submirror.

The root file system may not be mounted on a concat with more than one stripe.

You have tried to drop the slice that contains the root file system into a Concat/Stripe template. Remove one of the existing stripes.

The root file system may not be mounted on a trans device.

You have tried to drop the slice that contains the root file system into a Trans device template. The root file system cannot be placed in a Trans device.

Trans *dn* has no master

You have tried to commit a Trans device that has no master device. Add the master device and commit the device.

You cannot add a mounted slice to a mirror with multiple submirrors.

You have tried to attach a Concat/Stripe that has a mounted file system as a submirror and the Mirror already has a submirror that contains a mounted file system.

You cannot add *device* to a RAID device while it is initializing

You have tried to add a slice or hot spare pool to a RAID device that has been committed and is initializing. Wait until the device is initialized.

Dialog Box Warning Messages

DiskSuite Tool displays the following messages in Warning Dialog Boxes. When these messages are displayed you can either continue or undo the action. The information here will help you determine which action to select.

```
All of the state database replicas are on the same controller.
```

You have populated the MetaDB template with slices that are all attached to the same controller. If the controller fails, you will not have access to any of the metadevices.

```
Concat/Stripe dn has a slice slice with a mounted file
system. If an entry for its file system exists in
/etc/vfstab, it will be updated with the new device to mount for
the file system. You will need to reboot the system
to have the file system remounted.
```

You have tried to add a slice that contains a mounted file system to a Concat/Stripe template. The slice that contains the mounted file system must be the first slice added. You cannot add a mounted file system to a RAID device.

```
Metadevice device_type dn will be deleted.
Data could be lost.
Really delete it?
```

This message displays when you attempt to delete any committed metadevice. You should continue only if you are sure your data is protected.

Stripe component *dn* is not the same size as component *dn*. Extra space on the larger component will be wasted.

You have tried to add slices to a Concat/Stripe (stripe) that are a different size than the slices already in the stripe. Adding slices of a different sizes to a stripe causes wasted space.

Slice *dn* is on the same controller as slice *dn*. It is not advisable to have slices from multiple submirrors on the same controller.

You have tried to create a Mirror with submirrors that are made up of slices attached to the same controller. If the controller fails, the mirror will not protect against lost data.

Slice *dn* is on the same disk as slice *dn*. It is not advisable to have slices from multiple submirrors on the same disk.

You have tried to create a Mirror with submirrors that are made up of slices from the same disk. If the disk fails, the mirror will not protect against lost data.

Submirror *dn* is not the same size as submirror *dn*. Extra space on the larger submirror will be wasted.

You have tried to create a Mirror that has differently sized submirrors. The extra space on the larger submirror cannot be used.

Submirror `dn` has an erred component.
Its data will not be valid after it is detached.

You have tried to detach or offline a submirror that has a slice reporting errors.

The file system mounted on metadvice `dn` has
been unmounted since the last status update.

You have tried to delete a metadvice that unmounted. The device does not display the unmounted information because the Update Status button in the button panel of the Metadvice Editor window has not been pressed.

The following components are in the erred state: `dn`
You may not replace RAID component `dn` until they are fixed.

You are replacing a component of a RAID metadvice that has reported errors (in the last-errored state). This cannot be performed if there are any other components in the RAID metadvice that have reported errors.

The following components are in the erred state: `dn`
The data for the component replacing RAID component `dn` may be compromised.

You are replacing or enabling a RAID component that has reported errors. This action is dangerous if there is another component that has reported errors (in the last-errored state). The data on the new component may not be completely accurate.

```
The following components are in the last_erred state: dn
The data for RAID component dn may be compromised.
```

You are replacing or enabling a RAID component that is reporting errors. This action is dangerous if there is another component that has reported errors (in the last-errored state). The data on the new component may not be completely accurate.

```
The following components have erred: dn
The data for RAID component dn WILL NOT BE RESYNCD.
```

You have tried to replace a component in a RAID metadvice and there are two or more components reporting errors. It is not possible to replace the components because there is no way to recreate the data. If you proceed with the replacement, you must obtain the data from a backup copy.

```
The format of disk dn has changed.
You must restart metatool to incorporate the changes.
```

You have reformatted a disk that used to have a metadvice, file system, or database replica and pressed the Update Status button on the button panel of the Metadvice Editor window. If the disk is not being used, the new information is read by DiskSuite Tool and displayed in the appropriate windows (for example, Slice Browser and Disk View).

The log device for Trans **dn** cannot be detached until the Trans is unmounted or the system is rebooted.

You have tried to detach a log and commit the Trans object. The detach will not be performed as long as the log master is mounted. The Trans device is actually in a “detach pending state.”

The master device **dn** has a slice with a mounted file system.

In order for logging of this file system to be activated, the file `/etc/vfstab` must be updated with the new device name and the system rebooted.

Committing Trans **dn** will update `/etc/vfstab` automatically if an entry exists for the file system.

You have tried to add a metadvice that has a mounted file system to a Trans master. DiskSuite Tool will automatically change the entry for the file system in the `/etc/vfstab` file. If an entry for the file system does not exist in the `/etc/vfstab` file, you must create one. The message also tells you to reboot the system.

The master device `dn` for Trans `dn`
has a mounted file system.

If an entry for its file system exists in `/etc/vfstab`, it will
be updated with the new device to mount for the file system.

The system must be rebooted for this
device mount to take effect.

You have tried to add a master device that has a mounted file system to a Trans. DiskSuite Tool will automatically change the entry for the file system in the `/etc/vfstab` file. If an entry for the file system does not exist in the `/etc/vfstab` file, you must create one. The message also tells you to reboot the system.

The metadvice `dn` has been removed as a swap device
since the last status update.

You have tried to delete a device that is the swap device. The device does not say it is swap because the Update Status button in the button panel of the Metadvice Editor window has not been pressed.

The new Mirror device `dn`
has a submirror with a mounted file system.

If an entry for its file system exists in `/etc/vfstab`, it will
be updated with the new device to mount for the file system.

The system must be rebooted for this
device mount to take effect.

You have tried to add a Concat/Stripe that has a mounted file system to a Mirror. DiskSuite Tool will automatically change the entry for the file system in the `/etc/vfstab` file. If an entry for the file system does not exist in the `/etc/vfstab` file, you must create one. The message also tells you to reboot the system.

The new Concat/Stripe device `dn`
has a slice with a mounted file system.

If an entry for its file system exists in `/etc/vfstab`
it will be updated when the Concat/Stripe is committed so that
the next mount of the file system will use the new device.

The system must be rebooted for this
device mount to take effect.

You have tried to add a slice that has a mounted file system to a Concat/Stripe. DiskSuite Tool will automatically change the entry for the file system in the `/etc/vfstab` file. If an entry for the file system does not exist in the `/etc/vfstab` file, you must create one. The message also tells you to reboot the system.

The state database will have no replicas.
If the system reboots, all metadevices will
be corrupted.

You have tried to remove the state database and all replicas from the MetaDB template. If you commit, you will not have access to any metadevices.

The submirror *dn* has a slice with a mounted
file system.

In order for mirroring of
this file system to be activated, the file
/etc/vfstab must be updated with the
new device name and the system rebooted.

Committing Mirror *dn* will update
/etc/vfstab automatically if an entry exists
for the file system.

You have tried to add a submirror that has a mounted file system to a Mirror. DiskSuite Tool will automatically change the entry for the file system in the /etc/vfstab file. If an entry for the file system does not exist in the /etc/vfstab file, you must create one. The message also tells you to reboot the system.

This log is not mirrored. It is recommended that you mirror logs
whenever possible to avoid single points of failure.

You have tried to create a Trans device with a log that is not mirrored. If the log is not mirrored, the data could be lost or unavailable.


```
Trans dn has no log device.
```

You have tried to commit a Trans device that has no Trans log. You should add the log before committing the device. Until you add the log, the logging feature is disabled.

Dialog Box Information Messages

DiskSuite Tool displays the following messages in information dialog boxes. These messages tell you a common mistake has been made or provide a helpful message. These dialog boxes appear with a large “i” on the left side of the message. There is only one button on the bottom of this dialog: OK.

```
You added additional capacity to the metadvice metadvice_type on  
which file system file_system is mounted. You have the  
option of growing this file system to take advantage of  
this additional space either now or later (by hand).  
If you choose to grow it now, the application will be  
disabled until the growth process completes.
```

```
The command that will be run is:
```

```
"growfs -M file_system /dev/md/rdsk/dn"
```

```
Do you want to grow the metadvice now or later?
```

You have tried to add slices to a Concat/Stripe metadvice. Following a commit, you can expand the file system, as documented in Chapter 11, “Expanding a File System.”

```
The file system file_system mounted on metadvice dn  
is now being grown.
```

You are growing a file system.

Metadevice Editor Window Messages

The following messages are displayed when you are pointing to an object inside the Metadevice Editor window. An explanation of the message follows the sample output.

```
Click or drag to create a new object device
```

You are pointing at any of the five Template icons. *object* is either Trans, RAID, Mirror, Concat/Stripe, or Hot Spare Pool.

```
component_type dn: size=size, use=use, status=status
```

You are pointing at an object (*component_type*) on the canvas. The *component_type* is either Trans, RAID, Mirror, Concat/Stripe, or Hot Spare. The metadevice name is reported as *dn*, where the default size for *n* is a number in the range 0 to 127. The *size* is the capacity of the metadevice (for example, 500 Mbytes). The *use* is either Unassigned, Submirror, or */filesystem*. The *status* is reported as OK, Attention, Urgent, or Critical.

```
Drag objects into this work area to look at or modify them...
```

You are pointing at an empty canvas or the device list in the Metadevice Editor window.

```
hspnnn: status=status
```

You are pointing at a Hot Spare Pool on the canvas. The Hot Spare Pool name is reported as hspnnn, where *nnn* is a number in the range 000 to 999. The *status* is reported as OK, Attention, Urgent, or Critical.

```
Slice cntndnsn: size=size, use=use, status=status
```

You are pointing at a disk slice on the canvas. The name of the slice appears in the format, *cntndnsn*. The *size* is the capacity of the slice (for example, 5 Mbytes). The *use* is either Unassigned or Component. The *status* is reported as OK, Attention, Urgent, or Critical.

```
Use Button2 to pan the viewport over the work area ...
```

You are pointing at the panner. By pressing the middle button and moving the cursor, you move the canvas to a new view area.

Messages Displayed When Dragging an Object

The following messages are displayed when you are dragging an object inside the Metadevice Editor window. An explanation of the message follows the sample output.

```
Concatenations must be part of a mirror  
for hot spare pools to function
```

You are dragging a Hot Spare Pool over a concatenation. This message is telling you that the Concat/Stripe must be part of a Mirror or the Hot Spare Pool you are dropping will not work.

```
Drop a concatenation into mirror dn
```

You are dragging a Concat/Stripe over the specified Mirror. If you drop the Concat/Stripe, it will become part of that Mirror.

Drop a concatenation to replace submirror *dn*

You are dragging a Concat/Stripe over the specified submirror. Drop the Concat/Stripe inside the rectangle that contains the submirror to make the replacement.

Drop a hot spare pool to assign it to concatenation *dn*

You are dragging a Hot Spare Pool over the specified concatenation. By dropping the Hot Spare Pool into the Concat/Stripe, it becomes associated with that concatenation.

Drop here to associate a hot spare pool with this RAID

You are dragging a Hot Spare Pool over the specified RAID device. If you drop the Hot Spare Pool, it will become associated with the RAID device.

Drop a metadvice or slice into trans log

You are dragging a metadvice or slice over a Trans device. If you drop the metadvice or slice, it will become part of the Trans device.

Drop a metadvice or slice into master

You are dragging a metadvice or slice over the Master of a Trans device. Drop the object into the Master to add it to the device.

Drop a slice into hot spare pool *dn*

You are dragging a slice over the specified Hot Spare Pool. Drop the slice to add it to the Hot Spare Pool.

Drop a slice into RAID *dn*

You are dragging an unused slice over the specified RAID device. If you drop the slice, it will become part of the RAID device.

Drop a slice of the same size to replace the current slice

You are dragging a slice either over a committed RAID device or over a submirror that has more than one submirror. You can drop the new slice on the existing slice to make a replacement.

Drop a slice to replace current slice

You are dragging an unused slice over a Concat/Stripe, RAID, or Trans device. To replace the slice you are over, release the middle button and drop the slice.

Drop a slice into stripe *dn* or commit

You are dragging an unused slice over a Concat/Stripe that has one or more slices. You can populate the Concat/Stripe with additional slices or select the Concat/Stripe (stripe) and execute a commit.

Drop a slice to add new replicas; you should have at least three replicas.

You are dragging a slice over the MetaDB object. Drop the slice to create another replica. DiskSuite requires the configuration have a minimum of three slices in the MetaDB object.

Drop at least one concatenation into mirror *dn*

You are dragging a Concat/Stripe over the specified Mirror. You must drop a minimum of one Concat/Stripe into the specified Mirror.

Drop at least one slice into stripe *dn*

You are dragging an unused slice over a Concat/Stripe that has zero slices. You must populate the Concat/Stripe (stripe) with a minimum of one slice.

Drop at least three to create the RAID parity group

You are dragging an unused slice over the specified RAID device. You must drop a minimum of three slices into the RAID device.

Drop a slice to add a new replica.

You are dragging a slice over the MetaDB object. Drop the slice to create another replica.

You cannot add more concatenations;
mirror **dn** already has three submirrors

You are dragging a Concat/Stripe over the specified Mirror. You cannot add another Concat/Stripe (submirror) to a Mirror that already has three submirrors.

You cannot add slices to committed stripe **dn**

You are dragging an unused slice over a committed Concat/Stripe. DiskSuite Tool does not permit you to add slices to a committed Concat/Stripe (stripe).

You cannot replace in-use slices in a hot spare pool

You are dragging an unused slice over a slice that is in use in a committed Hot Spare Pool. You cannot drop the new slice on a slice in a Hot Spare Pool that is currently in use.

You cannot replace objects in committed RAID **dn**

You are dragging an object over the specified RAID device. Because the device is committed, you cannot make replacements.

You cannot replace slices in a committed stripe unless it is part of a submirror

You are dragging a slice over a a committed Concat/Stripe. You cannot make changes to this metadvice, unless it is part of a submirror.

You cannot replace submirror *dn* when mirror *dn* has only one submirror

You are dragging a submirror over the specified submirror. You cannot drop the submirror into the Mirror when there is only one submirror present.

You cannot replace slices in a committed trans device

You are dragging an unused slice over a slice that is in use as a Trans master or log. You cannot replace slices in a committed Trans device.

Disk View Window Messages

The Disk View window has a message line at the bottom that displays information about objects and actions. There are two types of message line feedback displayed:

- When you are pointing the cursor at an object, the message line has the following format:

object_type object_name: attribute=value, . . .

If you are pointing at a disk or slice that has a status problem, the message has the form:

```
object_type object_name: problem_description, affected_device
```

- When you are pointing the cursor at an empty portion of the canvas, the following message displays:

```
Drop object onto color drop sites to show mappings
```

You can select a disk or slice and drag it to the color map at the bottom of the Disk View window. On a color monitor, you have four colors available as drop sites. On a monochrome monitor, you have one color drop site.

Messages Displayed When Pointing at an Object

The following messages are displayed when you are pointing to an object inside the Disk View window:

```
Slice cntndnsn: size=size, use=use, status=status
```

You are pointing at a disk slice on the canvas. The name of the slice appears in the format, *cntndnsn*. The *size* is the capacity of the slice (for example, 5 Mbytes). The *use* is either Unassigned, Component, Hot Spare, MetaDB Replica, Reserve, *mount_point*, swap, Trans Log, or Overlap. The *status* is reported as OK, Attention, Urgent, or Critical.

Messages Displayed When Dragging an Object

The following messages are displayed when you are dragging an object inside the Disk View window.

```
Drag slices onto object templates in the metadvice editor canvas
```

You are dragging an object from the Disk View window. You can drop the slices inside an object or on the canvas of the Metadvice Editor window.

Log Messages

Log messages are those passed by `syslog(3)` to the `syslogd(1M)`. These messages are appended to a file and written to the console window. These messages will not appear in any DiskSuite Tool error or problem list.

The log messages are divided into the following categories:

- Notice log messages
- Warning log messages
- Panic log messages

The log messages displayed by DiskSuite are listed in alphabetical order below. Each message is always preceded with `md:`. The variables in these messages indicate the following:

- *dev* is a device name.
- *dnum* is a metadvice name.
- *num* is a number.
- *state* is a Trans device state
- *trans* is either “logging” or “master”

Note – When the initial portion of a message begins with a variable, the message is alphabetized by the first word following the variable.

Notice Log Messages

Could not load misc /*dev*

The named misc module is not loadable. It is possibly missing, or something else has been copied over it.

```
db: Parsing error on `dev`
```

The set command in `/etc/system` for the `mddb.bootlist<number>` is not in the correct format. Run `metadb -p` to place the correct set commands into the `/etc/system` file.

```
dnum: Hotspared device dev with dev
```

The first device name listed has been hot spare replaced with the second device name listed.

```
dnum: Hotspared device dev(num, num) with dev(num, num)
```

The first device number listed has been hot spare replaced with the second device number listed.

```
dnum: no mem for property dev
```

Memory could not be allocated in the `prop_op` entry point.

Warning Log Messages

```
dnum: Cannot load dev driver
```

The underlying named driver module is not loadable (for example, `sd`, `id`, or a third-party driver). This could indicate that the driver module has been removed.

```
Open error of hotspare dev  
Open error of hotspare dev(num, num)
```

The named hot spare cannot be opened, or the underlying driver is not loadable.

```
dnum: read error on dev  
dnum: write error on dev
```

A read or write error has occurred on the specified metadvice at the specified device name. This happens if any read or write errors occur on a metadvice.

```
dnum: read error on dev(num, num)  
dnum: write error on dev(num, num)
```

A read or write error has occurred on the specified metadvice at the specified device number. This happens if any read or write errors occur on a metadvice.

```
dnum: read error on dnum  
dnum: write error on dnum
```

A read or write error has occurred on the specified metadvice at the specified device number. This happens if any read or write errors occur on a metadvice.

```
State database commit failed
State database delete failed
```

These messages occur when there have been device errors on components where the state database replicas reside. These errors only occur when more than half of the replicas have had errors returned to them. For example, if you have three components with state database replicas and two of the components report errors, these errors may occur. The state database commit or delete is retried periodically. If a replica is added, the commit or delete will finish and the system will be operational. Otherwise, the system will time out and panic.

```
State database is stale
```

This message occurs when there are not enough usable replicas for the state database to be able to update records in the database. All accesses to the metadvice driver will fail. To fix this problem, add more replicas or delete inaccessible replicas.

```
trans device: read error on dnum
trans device: write error on dnum
```

A read or write error has occurred on the specified logging or master device at the specified metadvice. This happens if any read or write errors occur on a logging or master device.

```
trans device: read error on dev
trans device: write error on dev
```

A read or write error has occurred on the specified logging or master device at the specified device name. This happens if any read or write errors occur on a logging or master device.

```
trans device: read error on dev(num, num)
trans device: write error on dev(num, num)
```

A read or write error has occurred on the specified logging or master device at the specified device number. This happens if any read or write errors occur on a logging or master device.

```
logging device: dnum changed state to state
logging device: dev changed state to state
logging device: dev(num, num) changed state to state
```

The logging device and its associated master device(s) have changed to the specified state(s).

Panic Log Messages

```
State database problem
```

A failed metadvice state database commit or deletion has been retried the default 100 times.

```
dnum: Unknown close type  
dnum: Unknown open type
```

A metadvice is being opened/closed with an unknown open type (OTYP).

Man Page



This appendix contains the `metatool(1M)` man page.

Note – The man page is included only in the printed version of this document. The man page is not available when you use the AnswerBook online documentation to view this manual.



Index

A

- adding hot spares, 176
- AnswerBook, 24
- attach logging device
 - definition, 4
- attaching submirrors, 126
 - definition, 4

B

- block address
 - range, 60
- booting
 - from alternate root file system, 120
- browsers
 - accessing objects, 51
 - overview, 46
 - setting filters, 50
- button
 - definition, 4

C

- cascading menu
 - definition, 4
- checking status of
 - concatenations, 91
 - hot spares, 191

- mirrors, 141
- RAID, 217
 - state database replicas, 234
 - stripes, 91
 - trans devices, 165
- clearing metadevices, 84
- component
 - definition, 4
 - replacing within a submirror, 134
- concatenated stripe, 74 to 79
 - definition, 4
- concatenation
 - clearing, 84
 - compared to striping, 61
 - conceptual overview, 59 to 61
 - definition, 4
 - of metadevices, 63 to 67
 - of stripes, 74 to 79
 - overview, 27
- configuration guidelines, 57
- configuration log window
 - overview, 53
- confirmation dialog box, 53
- considerations
 - hardware and software, 63
- creating replicas, 225
- customer service, 3

D

- definition of terms, 4 to 7
- deleting
 - hot spares, 179 to 180
- detach submirror
 - definition, 4
- detaching logging device
 - definition, 4
- detaching submirrors, 4, 126
- dialog box
 - definition, 4
 - error messages, 267 to 270
 - information messages, 279
 - warning messages, 271 to 279
- dialog boxes
 - overview, 52
- disk drives supported, 3
- disk view window
 - overview, 38
- disk view window messages, 286 to 287
- diskset
 - conceptual overview, 193 to 194
 - definition, 5
 - overview, 28
- DiskSuite Tool
 - interfacing with, 29
- drag and drop
 - definition, 5

E

- error dialog box, 53
- expanding disk space, 236
- expanding file systems, 29

F

- failed boots
 - recovering from, 56
- file system
 - expansion overview, 235 to 236
 - nonexpandable, 236
- functionality provided, 2

G

- getting help, 3
- growing file systems
 - conceptual overview, 235 to 236
 - overview, 29

H

- help
 - accessing, 55
 - finding information, 56
 - overview, 55
- highlight
 - definition, 5
- hot spare
 - adding, 176
 - conceptual overview, 167 to 168
 - conditions to avoid, 168
 - defining, 168 to 171
 - definition, 5
 - enabling, 183
 - replacing, 180 to 183
- hot spare pool
 - and adding hot spares, 176
 - and deleting hot spares, 179 to 180
 - and removing hot spares, 179 to 180
 - definition, 5
 - naming conventions, 168
- hot spare pool browser, 48
- hot spares
 - overview, 28

I

- icon
 - definition, 5
- information dialog box, 53
- information screens
 - concat, 87
 - disk, 43
 - hot spares, 188
 - metadevice database replicas, 231
 - mirror, 137
 - overview, 42

-
- RAID, 214
 - slice, 44
 - stripe, 89
 - trans, 163
 - installation
 - basic steps, 10
 - interlace
 - changing the value on RAIDs, 202
 - changing the value on stripes, 68, 71, 77, 90, 216
 - default value, 60
 - definition, 5
 - specifying, 60
 - interlace value
 - default, 62
 - invoking DiskSuite Tool, 30
- L**
- log messages, 288 to 293
 - and types, 288
 - notice, 289
 - panic, 293
 - warning, 290 to 292
 - logging
 - and log location, 146
 - conceptual overview, 144
 - definition, 5
 - how it works, 144
 - overview, 27
 - removing, 154 to 156
 - logging device
 - and space required, 145
 - definition, 5
 - detaching, 154 to 156
 - shared, 144, 150 to 154
- M**
- MANPATH variable, 23
 - master device
 - definition, 5
 - menu bar
 - definition, 5
 - metadevice
 - and file system commands, 62
 - as concatenated stripes, 74 to 79
 - as logging devices, 162
 - clearing, 84
 - concatenated, 63 to 67
 - definition, 5
 - overview, 26
 - striped, 68 to 73
 - used as a raw device, 62
 - metadevice browser, 47
 - metadevice editor window
 - overview, 32
 - metadevice editor window messages, 280 to 286
 - metadevice state database
 - definition, 5
 - metadriver
 - definition, 6
 - metamirror
 - and failed component replacement, 167
 - mirror
 - See also* submirror
 - as logging devices, 162
 - definition, 6
 - mirroring
 - and online backups, 137
 - components on the same disk, 94
 - conceptual overview, 94
 - definition, 6
 - existing file systems, 100
 - file systems you cannot umount, 110
 - naming of metadevices, 94
 - overview, 27
 - unused slices, 95
 - mirroring options
 - pass (resync), 139
 - reads, 140
 - writes, 140
 - mirrors
 - checking status, 141
 - creating, 95

N

- naming conventions
 - for metatrans devices, 144
- newfs command, 27
- nonexpandable file systems, 236
- notice log messages, 289

O

- online backups, 137
- optimized resync
 - definition, 6

P

- panic log messages, 293
- partial resync
 - definition, 6
- partition
 - used for replicas, 221
- path requirements, 23
- PATH variable, 23
- peripherals supported, 3
- pkgadd command, 12
- pkginfo command, 21 to 22
- pkgrm command, 22
- popup menu
 - definition, 6
- problem list window
 - overview, 54
- pull down menu
 - definition, 6

R

- RAID
 - assigning hot spare pools, 208
 - concatenating components, 204
 - conceptual overview, 197
 - creating RAID metadevices, 199
 - definition, 6
 - hardware and software
 - considerations, 199

- information screen, 214
- operations supported, 198
- replacing components, 210

RAID devices

- overview, 28

- reconfiguring submirrors, 121

- recovery from failed boots, 56

- related documentation, xvii

removing

- hot spares, 179 to 180
- replicas, 228

- removing Online: DiskSuite product, 22

replacing

- submirror components, 134

replica

- definition, 6

replicas

- and planning locations, 221
- and reserved space, 219
- conceptual overview, 219 to 221
- creating, 225
- minimum number, 221
- removing, 228

resync region

- definition, 6

resyncing

- definition, 6

S

selection button

- definition, 6

- shareall command, 11

- sharing logging devices, 150 to 154

- slice browser, 47

- slice, *See* partition

state database

- and adding replicas, 225
- and removing replicas, 228
- and reserved space, 219
- conceptual overview, 219 to 221
- definition, 6

- state database replicas

- overview, 28
- status of
 - concatenations, 91
 - hot spares, 191
 - mirrors, 141
 - RAID, 217
 - state database replicas, 234
 - stripes, 91
 - trans devices, 165
- stripe
 - clearing, 84
 - concatenated, 74 to 79
 - definition, 7
- striped metadevices, 68 to 73
- striping
 - compared to concatenation, 61
 - conceptual overview, 59 to 61
 - metadevices, 68 to 73
 - overview, 27
- submirror
 - See also* mirror
 - and file system expansion, 236
 - and hot spare pools, 186 to 188
 - and replacing components, 134
 - definition, 7
 - detaching, 4
 - overview, 27
- supported platforms, 3

T

- technical support, 3
- trans device
 - and file system commands, 144
 - creating, 146 to 150
 - definition, 7

U

- UFS
 - definition, 7
- UFS logging
 - and log location, 146
 - and shared devices, 150 to 154
 - conceptual overview, 144

- definition, 7
- overview, 27
- removing, 154 to 156
- unmirroring
 - file systems, 108
 - file systems you cannot umount, 120

W

- warning dialog box, 53
- warning log messages, 290 to 292

