



Solaris Handbook for Sun™ Peripherals

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A

Part No.: 805-7404-10
March 1999, Revision A

Send comments about this document to:
docfeedback@sun.com

1999 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

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

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

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

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

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

Copyright 1998 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303 Etats-Unis. Tous droits réservés.

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

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

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

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

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPENDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Please
Recycle



Adobe PostScript

Contents

- 1. Before You Start 1**
 - Requirements 1
 - Peripheral Installation Overview 1
 - Shut Down the System 3
 - Run Levels 5
 - The `shutdown` Command 7

- 2. Setting Up a Disk Drive 11**
 - Requirements 11
 - Disk Drives 12
 - Preparing the Disk 13
 - The `format` Utility 13
 - Creating and Mounting File Systems 21

- 3. Setting Up and Using a Tape Drive 27**
 - Requirements 27
 - Tape Drives 28
 - Miscellaneous Tape Commands 30
 - Tape Utilities 32
 - `cpio` Command 33

dd Command	34
ufsdump Command	34
ufsrestore Command	35
tar Command	37

4. Setting Up Removable Media Devices 39

Requirements	39
CD-ROM Drives	40
Diskette Drives	42
PCMCIA Memory Cards	42
How Solaris Manages CD-ROM and Diskette Devices	44
Volume Management	45
Comparing Automatic and Manual Mounting	47
Reference Material for CD-ROM and Diskette Devices	49
Using CDs and Diskettes	51

5. Setting Up an Option Card 53

Option Cards	53
Requirements	54
To Add an Option Card	54

A. SCSI Addressing 57

SCSI Subsystem	57
SCSI Controller	58
SCSI Address Selection Schemes	59

B. Booting Your System 65

Tables

TABLE 1-1	SunOS and Corresponding Solaris Releases	2
TABLE 1-2	Shutdown Commands	4
TABLE 1-3	Run Levels	5
TABLE 2-1	<code>fields</code> <code>/etc/vfstab</code>	23
TABLE 3-1	Address and Logical Device Names for Tape Drives	28
TABLE 4-1	<code>fdformat</code> Utility Options	43
TABLE 4-2	<code>fdformat</code> Utility Device Name Options	44
TABLE 4-3	Device Control Syntax Descriptions	47
TABLE 4-4	CD-ROM and Diskette File System Mount Points	49
TABLE 4-5	CD-ROM and Diskette Device Locations in <code>/vol</code> With No File System Present	50
TABLE 4-6	What You Can Do With CDs and Diskettes	51
TABLE A-1	SCSI Target Addresses	60
TABLE A-2	Tape Drive Logical Device Names	63
TABLE B-1	Interactive Boot Procedure Steps	68

Preface

The *Solaris Handbook for SMCC Peripherals* describes how to add and configure additional Sun Microsystems peripheral devices, such as disk drives, tape drives, CD-ROM drives, and option cards to an existing system using the Solaris™ operating environment.

Note – The Solaris operating environment includes the SunOS™ operating system, the OpenWindows™ operating environment, the Common Desktop Environment (CDE), compilers, and other software.

Note – The instructions in this manual are designed for an experienced system administrator.

Before You Read This Book

Before you begin using the information in this book, you should:

- Read the documentation that accompanies your peripheral devices.
- Install the Solaris software on your system.

Note – If you have not yet installed the Solaris operating environment on your system, do so now.

Tip – For more information about how to install the Solaris operating environment on your system, refer to the *Solaris Advanced Installation Guide*.

How This Book Is Organized

This book contains the following chapters:

Chapter 1 describes the procedures that you must perform before you install new peripherals.

Chapter 2 describes how to configure disk drive devices using the Solaris operating environment.

Chapter 3 describes how to configure and use tape drive devices with the Solaris operating environment.

Chapter 4 describes how to configure CD-ROM drives, diskette drives and PCMCIA memory cards.

Chapter 5 describes how to configure an option card using the Solaris software environment.

Appendix A describes addressing concepts for the SCSI peripheral subsystem.

Appendix B describes how to boot your system when you add a new device.

Using UNIX Commands

This document does not contain information on basic UNIX[®] commands.

See one or more of the following for this information:

- AnswerBook[™] online documentation for the Solaris software environment
- Other software documentation that you received with your system

Typographic Conventions

TABLE P-1 Typographic Conventions

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

Shell Prompts

TABLE P-2 Shell Prompts

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

Related Documentation

TABLE P-3 Related Documentation

Application	Title	Part Number
System administration	<i>Solaris 1.x (SunOS 4.x) Handbook for SMCC Peripherals</i>	801-2424
System administration	<i>System Administration Guide, Vol. I</i>	805-3727
System administration	<i>System Administration Guide, Vol. II</i>	805-3728
System administration	<i>OpenBoot 2.x Command Reference Manual</i>	805-4434
System administration	<i>OpenBoot 3.x Command Reference Manual</i>	805-4436

Sun Documentation on the Web

The `docs.sun.com` web site enables you to access Sun technical documentation on the World Wide Web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at <http://docs.sun.com>.

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at docfeedback@sun.com. Please include the part number of your document in the subject line of your email.

Before You Start

This chapter describes the procedures that you must perform before you install new peripherals.

Requirements

Before you continue with the procedures in this chapter you must have:

- The Solaris software installed
- Superuser privileges
- Booted your system

Note – For Solaris™ installation instructions and superuser information, refer to the *Solaris Advanced Installation Guide*.

Note – For information on booting your system, refer to “Booting Your System” on page 65 in the appendix.

Peripheral Installation Overview

The table below provides a list of the common steps you perform when you add a peripheral to a system. For more detailed information on a particular step refer to the specific section this book.

- “To Prepare the System” on page 2

- “Shut Down the System” on page 3
- Address and connect the peripheral (device specific)
- “Boot Your System”
- Use Software Utilities to Prepare the Peripheral for Use (device specific)

▼ To Prepare the System

1. Verify the revision of the operating system with the `uname -rs` command:

Type `uname -rs`:

```
# uname -rs
SunOS 5.x
```

The SunOS revision is displayed.

The *Solaris system environment*, to which this manual refers, includes the SunOS operating system as well as compilers and other software. To determine the Solaris system environment that corresponds with your SunOS see TABLE 1-1 on page 2.

TABLE 1-1 shows which SunOS revisions were supplied with the various releases of the Solaris operating environment.

TABLE 1-1 SunOS and Corresponding Solaris Releases

If you are running the following SunOS:	You are using the following Solaris system environment:
SunOS 5.0	Solaris 2.0
SunOS 5.1	Solaris 2.1
SunOS 5.2	Solaris 2.2
SunOS 5.3	Solaris 2.3
SunOS 5.4	Solaris 2.4
SunOS 5.5	Solaris 2.5
SunOS 5.5.1	Solaris 2.5.1
SunOS 5.6	Solaris 2.6
SunOS 5.7	Solaris 7

Note – If you are working in an environment other than Solaris 2.x or Solaris 7, the commands and tasks described in this manual do not apply.

Note – If you are using SunOS 4.x, refer to the *Solaris 1.x (SunOS 4.x) Handbook for SMCC Peripherals*.

2. Create the `/reconfigure` file:

```
# touch /reconfigure
```

The `touch /reconfigure` command performs the same function as the `boot -r` command. It allows the operating system to recognize the presence of all newly installed peripheral devices once the system is rebooted.

3. Add a device driver, if necessary.

If a new device driver is required, use the `pkgadd` command or the Software Manager tool to perform this task.

Note – This task is optional; most device drivers are already part of the operating system. If a device driver is required, you would typically find a CD-ROM packaged with the new peripheral, as well as instructions on how to add the device driver. If you cannot find a CD-ROM or instructions on how to add a specific device driver, a new device driver is probably not needed. Therefore, you can skip this step.

Note – Refer to the *System Administration Guide* and the *Solaris Advanced Installation Guide* for instructions on adding a device driver.

Shut Down the System

Unless you are adding a *hot-pluggable* peripheral, you must shut down the operating system and power off the system before the new peripheral is installed.

Before you shut down the operating system, you should stop any applications and close any open files on the system.

When preparing to shut down a system, you need to determine which of the following shutdown commands is appropriate for the system and the task at hand:

- /sbin/init
- /usr/sbin/shutdown
- /usr/sbin/halt
- /usr/sbin/reboot

Note – See TABLE 1-2 on page 4 for a definition of each shutdown command.

These commands initiate shut down procedures, kill all running processes, write data to disk, and shut down the system software to the appropriate run level.

Note – `init` and `shutdown` are the two most reliable ways to shut down a system because they use `rc` scripts to kill running processes and shut down the system with minimal data loss. The `halt` and `reboot` commands do not run the `rc` scripts properly and are not the preferred method for shutting down the system.

Note – For more information about `rc` scripts, see the *System Administration Guide*, “Boot Files and Run Levels.”

The table below describes each of the shutdown *commands*. For shutdown *procedures* refer to “To Shut Down a Single-User System Using `init`” on page 6 and “To Shut Down a Multiuser System Using `shutdown`” on page 7.

TABLE 1-2 Shutdown Commands

Command	Function
<code>init n</code>	Use the <code>init</code> command to shut down a system when there are no users who need to be warned. <code>init</code> can also be used to change run levels. For example, use <code>init 6</code> to reboot a system. See “Run Levels” on page 5” for more information.
<code>shutdown</code>	Use the <code>shutdown</code> command when shutting down a system with multiple users. The <code>shutdown</code> command sends a warning message to all users who are logged in, waits for 60 seconds (the default), and then shuts down the system to single-user state. See the procedure shown on page 7.
<code>halt</code>	Use the <code>halt</code> command when the system must be stopped immediately and it is acceptable not to warn any current users. The <code>halt</code> command shuts down the system without any delay. It does not warn any other users on the system. It does not run any customized shutdown scripts from the <code>rc</code> directories.
<code>reboot</code>	Use the <code>reboot</code> command to shut down a single-user system and bring it into multiuser state. <code>reboot</code> does not warn other users on the system. It does not run any customized shutdown scripts from the <code>rc</code> directories.

Run Levels

The run level defines what services and resources are available to users. A system can be in only one run level at a time. See the table below for a description of the various run levels.

Run level numbers are used with the `init` and `shutdown` commands to control system services. Use the `init` command to shut down a system that does not have multiple users or clients. In addition, use the `init` command to change run levels. You can also specify the run level as an argument to the `shutdown` command with the `-i` option.

TABLE 1-3 Run Levels

Run Level	System State	Description
0	Monitor mode	The operating system is shut down. The system displays the <code>ok</code> prompt. It is safe to turn off system power.
3	Multuser mode	The operating system is fully booted; all required processes are started to support multiple users, network access, file systems, event logging, and so on. It is not safe to turn off system power.
5	Automatic power-off	The operating system is shut down and power is automatically turned off (on supported systems only).
6	Reboot	The operating system is brought to run level 0 and then to the default run level as specified in the <code>/etc/inittab</code> file (typically run level 3).
s or S or 1	Single-user mode	The operating system is in maintenance mode. In this mode, only a few processes are started for basic system functionality. Only the <code>/</code> , <code>/usr</code> , and <code>/proc</code> file systems are mounted, and no <code>nfs</code> file systems are shared. Only superuser is allowed to log in. It is not safe to turn off system power.

Use the `who` command to determine your run level.

The following example shows a system that is in multuser mode (run level 3).

```
# who -r
. run-level 3 Apr 24 10:44 S 0 S
```

Note – `who` is not a valid command when you are at run level 0.

▼ To Shut Down a Single-User System Using `init`

This section describes how to shut down a single-user system.

Note – `init` does not send a warning message to users.

Note – Be sure to prepare the system first. Refer to “To Prepare the System” on page 2 for more information.

1. **Become superuser.**
2. **Stop all applications and close any open files on the system.**
3. **Shut down the system.**

```
# init 0
```

This command writes data to the disk, kills all active processes, brings the system down to the monitor mode, and displays the `ok` prompt.

4. **If you are asked for confirmation, type `y`.**

```
ok Do you want to continue? (y or n): y
```

5. **If you are installing a new SCSI device continue with Step 6.**

If you are installing a new card continue with instructions in Chapter 5, “Setting Up an Option Card”.

6. **Reset the SCSI bus with the `reset` command followed by a Stop-A:**

```
ok reset  
Immediately press <Stop-a> once you see the banner on your screen.
```

The `reset` command (`reset-all` on some systems) resets the SCSI bus and memory to ensure an effective `probe-scsi` search. The Stop-A key sequence interrupts the reboot process that normally results from the `reset` command. This way you remain at run level 0 to perform the `probe-scsi` command.

Note – Be sure to turn on all SCSI devices, such as disk and tape drives, before you attempt to use the `probe-scsi` or `probe-scsi-all` commands. The `probe-scsi` command can detect connected SCSI devices only if they are turned on.

7. List the SCSI target IDs that are currently used on the SCSI bus by typing:

```
ok probe-scsi
```

If you have more than one SCSI bus on your system, use the `probe-scsi-all` command. This command probes for all SCSI devices on every SCSI bus.

8. Choose an unused SCSI target ID for your new device.

Note the target IDs that are already assigned and choose an unused target ID for your new device. For more information on selecting a target ID see Appendix A, “SCSI Addressing”.

9. After the `ok` prompt is displayed, turn off the power to the system and then to all external peripheral devices.

You are now ready to install the peripheral device. Refer to the documentation that accompanies your hardware and the appropriate section in this manual:

- “Setting Up a Disk Drive” on page 11
- “Setting Up and Using a Tape Drive” on page 27
- “Setting Up Removable Media Devices” on page 39

The shutdown Command

Use the `/usr/sbin/shutdown` command to shut down servers when users must be notified of the impending shutdown.

▼ To Shut Down a Multiuser System Using `shutdown`

This section describes how to shut down a multiuser system, such as a server.

1. Become superuser.

2. To find out if users are logged into the system, type:

```
# who
```

A list of all logged-in users is displayed. You can send email or broadcast a message to let users know that the system is being shut down.

3. Stop all applications and close any open files on the system.

4. Ensure that the operating system checks for newly installed peripheral devices when you power on or reboot your system, type:

```
# touch /reconfigure
```

The `touch /reconfigure` command performs the same function as the `boot -r` command. It allows the operating system to recognize the presence of all newly installed peripheral devices.

5. Shut down the system, by typing:

```
# /usr/sbin/shutdown -y -ggrace_period -irun_level
```

In this command:

<code>shutdown</code>	Brings a system to run level <i>s</i> (single-user mode) by default. A message is broadcast to all users. After a 60-second wait (default), you are asked to confirm that you want to shut down the system.
<code>-y</code>	Continues to shut down the system without intervention; otherwise, you are prompted to continue the shutdown process.
<code>-g<i>grace_period</i></code>	Indicates a time (in seconds) before the system is shut down. The default is 60 seconds. Specify a number for <i>grace_period</i> .
<code>-i<i>run_level</i></code>	Allows you to bring the system to a run level that is different from the default. (Refer to TABLE 1-3 on page 5 for run level information.) Specify a number for <i>run_level</i> .

In the following example, the system will be brought to run level 0 in 3 minutes without requiring additional confirmation.

```
# /usr/sbin/shutdown -y -g180 -i0
Shutdown started.      Fri Nov 4 11:07:33 MST 1994
Broadcast Message from root (console) on pluto Fri Nov 4 11:07:34
The system will be shut down in 3 minutes

      .
      .
      .
INIT: New run level: 0
The system is coming down. Please wait.

      .
      .
      .
The system is down.
syncing file systems... [11] [9] [5] done
Program terminated
Type help for more information
ok
```

6. If you are asked for confirmation, type `y`.

```
ok Do you want to continue? (y or n): y
```

7. If you are installing a new SCSI device continue with Step 8.

If you are installing a new card continue with instructions in Chapter 5, “Setting Up an Option Card.”

8. Reset the SCSI bus by typing:

```
ok reset
Immediately press Stop-A once you see the banner on your screen.
```

The `reset` command (`reset-all` on some systems) resets the SCSI bus and memory to ensure an effective `probe-scsi` search. The Stop-A key sequence interrupts the reboot process that normally results from the `reset` command. This way you remain at run level 0 to perform the `probe-scsi` command.

Note – Be sure to turn on all SCSI devices, such as disk and tape drives, before you attempt to use the `probe-scsi` or `probe-scsi-all` commands. The `probe-scsi` command can detect connected SCSI devices only if they are turned on.

9. List the SCSI target IDs that are currently used on the SCSI bus by typing:

```
ok probe-scsi
```

If you have more than one SCSI bus on your system, use the `probe-scsi-all` command. This command probes for all SCSI devices on every SCSI bus.

10. Choose an unused SCSI target ID for your device.

Note the target IDs that are already assigned and choose an unused target for your new device. For more information on selecting a target ID see Appendix A, “SCSI Addressing”.

11. After the `ok` prompt is displayed, turn off the power to the system and then to all external peripheral devices.

You are now ready to install the peripheral device. Refer to the documentation that accompanies your hardware and the appropriate section in this manual:

- “To Add a Disk Drive” on page 12
- “To Add a Tape Drive” on page 29
- “To Add a CD-ROM Drive” on page 40
- “To Add an Additional CD-ROM Drive” on page 41
- “To Add a Diskette Drive” on page 42

Setting Up a Disk Drive

This chapter describes how to configure disk drive devices using the Solaris™ operating environment.

This chapter contains the following information:

- “Requirements” on page 11
- “To Add a Disk Drive” on page 12
- “To Format a Disk” on page 14
- “To Partition a Disk” on page 16
- “To Label a Disk” on page 20
- “To Create a File System” on page 22
- “To Mount a File System” on page 22

Requirements

Before you configure the operating environment for a new disk drive, you must:

- Install the Solaris software.
- Shut down the system and peripherals that are already connected, as described in “Shut Down the System” on page 3.

Note – If you did not follow the procedures in “Shut Down the System” on page 3, the operating system may not recognize the new disk drive.

Disk Drives

Disk drives are common devices to add to a system. There are three major steps involved when adding a disk drive:

- Installing the disk (See “*To Add a Disk Drive*”).
- Preparing the disk with the `format` utility (See “*Preparing the Disk*”).
- Creating and mounting the file systems (See “*To Create a File System*”).

The last step may not be necessary if you plan to use special (Redundant Array of Inexpensive Disks) applications such as Veritas Volume Manager or Solstice DiskSuite™ software to manage the data on the disk.

▼ To Add a Disk Drive

1. Shut down your system.

See “Shut Down the System” on page 3.

2. Check that the address switch for the disk drive has been correctly set.

Each SCSI target ID determines the logical device name that the Solaris operating environment assigns to the drive. For more information about SCSI addressing refer to “SCSI Addressing” on page 57” in the appendix and to the *System Administration Guide*.



Caution – Each disk drive must have a unique SCSI target ID setting.

3. Install the disk drive.

For installation information, refer to the documentation that accompanies your hardware.

4. Turn the power on to the system and all the peripheral devices.

In most cases, this will cause the system to boot.

If the `/reconfigure` file is present (as described in “To Prepare the System” in Chapter 1), then the operating system automatically assigns a logical device name and initializes the appropriate device drivers for the new device. If the `/reconfigure` file does not exist, you can achieve the same effect by performing a `boot -r`. See Appendix B, “Booting Your System” for more information.

5. Prepare the disk, if necessary.

See “Preparing the Disk” on page 13.

Preparing the Disk

Before the a newly installed disk drive can be used it must be *formatted*, *partitioned* and *labeled*. Use the `format` utility to perform each of these activities as needed.

The `format` Utility

The `format` utility is a menu driven disk maintenance program that is used to format, partition, and label disk drives as well as manage defective sectors. When you start the `format` utility, it probes your system for disk drives, reads each label on each disk, and displays a disk list. Once you choose a disk to work with, `format` provides a menu with selectable routines.



Caution – Many routines in the `format` utility will overwrite data on disk drives. Always back up all of your data on *all* disk drives before using the `format` utility.

The `format` utility uses a disk statistic file called `/etc/format.dat` to determine default disk drive configuration parameters. However, `format` can automatically configure and label the SCSI disk drives, even if your specific type of drive is not described in the `/etc/format.dat` file. This lets you format, partition, and label any disk drive compliant with SCSI-2 without having to edit the `/etc/format.dat` file.

You must be superuser to use the `format` utility. If you are not superuser, you will see the following error message when you try to use `format`.

```
# format
Searching for disk...done
No permission (or no disk found)!
```

To use the `format` utility, refer to the following sections of this manual:

- “To Format a Disk” on page 14
- “To Partition a Disk” on page 16
- “To Label a Disk” on page 20

Note – For additional information about the `format` utility, refer to the *System Administration Guide*.

▼ To Format a Disk

When you format a disk you create disk sectors on the media and compile a list of defective sectors. The disk must be formatted before any data can be written on it. In most cases, disks are formatted by the manufacturer or reseller and do not need to be reformatted when you install the drive. To determine whether or not a disk is formatted, as well as to actually format a disk, use the `format` utility as described here.



Caution – Formatting a disk is destructive to data that might be on the disk.

Note – Not all menu information appears exactly as shown. If you have a SPARCstorage Array you must load the SPARCstorage™ Array drivers before using the `format` utility.

1. **Become superuser.**
2. **Start the `format` utility.**

```
# format
```

If the disks are already labeled, the system displays information similar to the following:

```
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t3d0 <SUN0207 cyl 1214 alt 2 hd 9 sec 36>
     /sbus@1,f8000000/esp@0,8000000/sd@3,0
  1. c0t0d0 <SUN0207 cyl 1214 alt 2 hd 9 sec 36>
     /sbus@1,f8000000/esp@0,8000000/sd@0,0
  2. c1t1d0 <SUN0207 cyl 1214 alt 2 hd 9 sec 36>
     /sbus@1,f8000000/esp@0,8000000/sd@1,0
  3. c1t2d0 <SUN0207 cyl 1214 alt 2 hd 9 sec 36>
     /sbus@1,f8000000/esp@0,8000000/sd@2,0
Specify disk (enter its number):2
```

In the above example:

- Disk drives 2 (`c1t1d0`) and 3 (`c1t2d0`) are newly installed disk drives.
- **2** is specified as the disk to work with at this time.

After you format the first new disk, format the second new disk the same way.

3. **Select the disk that you want to format from the list displayed on your screen.**



Caution – Make sure that you choose the correct disk. If you choose a disk that has data on it, that data will be overwritten.

The `format` utility will display a message indicating whether the disk is formatted or not:

```
Selecting c1t1d0
[disk formatted]
```

If your disk is *not* formatted, proceed to the Step 4.

If your disk *is* formatted, proceed to the *To Partition a Disk* procedure later in this chapter.



Caution – If you see the following message:

Warning: Current Disk has mounted partitions.

Do not proceed. Type `q` to exit the `format` utility. Go back to Step 2 and make sure that you are selecting the correct disk drive.

4. At the `format>` prompt, type `format`, and confirm the command by typing `y`.

```
format> format

Ready to format. Formatting cannot be interrupted.
Continue? y
Beginning format. The current time is Fri Sept 11 14:56:51 1993
Formatting ...
done

Verifying media ...
pass 0 - pattern = 0xc6dec6de
pass 1 - pattern = 0x6db6db6d
total of 0 defective blocks repaired.
format>
```

Format performs two operations:

- It formats the disk to write the sectors on the medium.
- It runs two verification passes to analyze the disk for possible defects and compiles a list of defective sectors for non-use.

It is not possible to predict how long formatting will take because it depends on many variables such as the size of the disk, your CPU and amount of memory, as well as other system activity. Times range from one minute to more than an hour.



Caution – Do not interrupt the formatting of a disk drive. An interruption at a critical moment could cause disk corruption.

▼ To Partition a Disk

Partitioning a disk involves dividing a disk into subsections called *partitions* or *slices*. Each individual partition appears to the operating system (and to the system administrator) as though it were a separate disk drive. File systems, swap partitions, and databases are created on partitions.

There are eight possible partitions represented by numbers 0 through 7. However, partition 2 is reserved as a single partition that always represents the entire disk drive. You can configure a disk with as many as seven individual partitions (numbered 0, 1, 3, 4, 5, 6, and 7 with partition 2 for the whole drive), or with as few as one single partition, or any combination between. How you choose to partition your disk depends on the plans that you have for your new disk. Consider how much data will be installed in a given partition, the rate that data will increase, and plan to have approximately 10% as free space that will never be used (there are exceptions to the 10% free space rule, see the *System Administration Guide* for more details).

Sometimes the manufacturer creates default partitions on the disk. However, due to the wide variety of storage space needs, the default partitions on the disk may not suit your needs. If this is the case, now is the time to re-partition the disk.



Caution – Partitioning a disk is destructive to data that might be on the disk.

1. **Become superuser and start the `format` utility.**

2. Type partition at the format prompt.

The following menu is displayed:

```
format> partition
PARTITION MENU:
    0      - change '0' partition
    1      - change '1' partition
    2      - change '2' partition
    3      - change '3' partition
    4      - change '4' partition
    5      - change '5' partition
    6      - change '6' partition
    7      - change '7' partition
select    - select a predefined table
modify    - modify a predefined partition table
name      - name the current table
print     - display the current table
label     - write partition map and label to the disk
!<cmd>   - execute <cmd>, then return
quit
partition>
```

3. Type modify.

The following menu is displayed.

```
Select partitioning base:
  0. Default partition for selected drive
  1. Current partition table (original sd3)
  2. All Free Hog
Choose base (enter number) [0]? 2
```

Note – Not all menu information appears exactly as shown. For disk drives greater than 2.1-Gbytes there is no default partitioning table available.

4. Select the number that corresponds to "All Free Hog" (2 in this example).

By choosing All Free Hog you will start the partitioning session with a blank partition table. All partition values (except partition 2) will be set to zero. This way no pre-existing partition values will interfere with your new partition values.

Once you have selected your partitioning base, the following table is displayed:

Part	Tag	Flag	Cylinders	Size	Blocks
0	-	-	0	0	(0/0/0)
1	-	-	0	0	(00/0/0)
2	-	-	0 - 1253	198.39MB	(1254/0/0)
3	-	-	0	0	(0/0/0)
4	-	-	0	0	(0/0/0)
5	-	-	0	0	(0/0/0)
6	-	-	0	0	(0/0/0)
7	-	-	0	0	(0/0/0)

These are your current partition table values. Partition 2 values represent the entire disk drive. Never attempt to change partition 2 values. In fact, this `modify` routine will not allow it.

5. Type `y` to continue:

```
Do you wish to continue creating a new partition
table based on above table[yes]? y
```

6. Select one of the seven partitions as a free hog partition:

```
Free Hog partition[6]? 6
```

When you use the `format` utility to change the size of one or more disk partitions, you must designate a temporary partition, the *free hog*, that can expand and shrink to accommodate resizing. This partition *frees* space when you expand a partition, and receives or *hogs* the discarded space when you shrink a partition. By the end of the `modify` session, the designated free hog partition must have space left in it and this partition will be a valid partition on your disk.

However, the free hog *designation* exists only during a given partitioning session. There is no permanent free hog partition during day-to-day operations.

Partition 6 is used in this example. *Do not* specify partition 2 as the free hog.

7. Specify the size of each partition in megabytes by changing any of the displayed sizes.

You are asked to specify the size of each partition. Type 0 to assign zero space to any unwanted partitions. Type a *size* followed by **mb** (for megabytes) for all desired partitions. You will not be asked for a size for partition 2 nor for the free hog partition (6 in this example).

```
Enter size of partition `0` [0b, 0c, 0.00mb]: 0
Enter size of partition `1` [0b, 0c, 0.00mb]: 0
Enter size of partition `3` [0b, 0c, 0.00mb]: 0
Enter size of partition `4` [0b, 0c, 0.00mb]: 0
Enter size of partition `5` [0b, 0c, 0.00mb]: 0
Enter size of partition `7` [0b, 0c, 0.00mb]: 80mb
```

The sizes have been changed as displayed:

Part	Tag	Flag	Cylinders	Size	Blocks
0	-	-	0	0	(0/0/0)
1	-	-	0	0	(00/0/0)
2	-	-	0 - 1253	198.39MB	(1254/0/0)
3	-	-	0	0	(0/0/0)
4	-	-	0	0	(0/0/0)
5	-	-	0	0	(0/0/0)
6	-	-	0 - 747	118.34MB	(748/0/0)
7	-	-	748 - 1253	80.05MB	(506/0/0)

In this example, the disk is set up to have two individual partitions; 6 and 7. Partition 2 is a special partition that represents the entire disk, sometimes referred to as the *overlap* partition.



Caution – Do not change partition 2. Reducing the size of partition 2 decreases available disk space.

8. Type *y* to confirm:

```
Okay to make this the current partition table [yes]? y
```

9. Type a name for the new partition table:

```
Enter table name (remember quotes): "table1"
```

This name is a temporary name used by the `format` utility to keep track of partition tables as they are created.

10. Type `y` to write the new partition table to the disk label:

```
Ready to label disk, continue? y
```

The label is where the partition table is stored. If you type `n` for no, you will lose the partition values that you just assigned and the previous partition table will remain in effect.

11. Type `q` to quit the partition routine, and another `q` to quit the format utility.

```
partition> q
.
.
.
format> q
```

▼ To Label a Disk

Labeling a disk involves using the `format` utility to write information to a special reserved area on a disk. This information is referred to as a disk label or VTOC (Volume Table of Contents). Each disk must have a disk label for the operating system to be able to use the disk.

A disk label contains two key pieces of information:

- Disk geometry (information that does *not* change)
- Disk partition table (information that *may* change)

A label is often created by the manufacturer, so you usually only need to label a disk when you change the partition information. If you use the `modify` routine as described in " *To Partition a Disk*", you will be asked to label your disk and you do not need to perform the steps listed in this section.

If the Solaris operating environment is unable to find a valid label on a drive, (as shown for `c1t2d0` below) the system displays information similar to the following:

```
# format
Searching for disks...done

c1t2d0:  configured with capacity of 198 MB
AVAILABLE DISK SELECTIONS:
  0.  c0t3d0 <SUN0207 cyl  1214 alt 2 hd 9 sec 36>
      /sbus@1,f8000000/esp@0,8000000/sd@3,0
  1.  c0t0d0 <SUN0207 cyl  1214 alt 2 hd 9 sec 36>
      /sbus@1,f8000000/esp@0,8000000/sd@0,0
  2.  c1t1d0 <SUN0207 cyl  1214 alt 2 hd 9 sec 36>
      /sbus@1,f8000000/esp@0,8000000/sd@1,0
  3.  c1t2d0 <Unable to read disk label>
      /sbus@1,f8000000/esp@0,8000000/sd@2,0
Specify disk (enter its number):3

Selecting c1t2d0
[disk formatted]
Disk not labeled. Label it now? y
```

The configured with capacity message and the Label it now? prompt are displayed for a disk without a valid label or for a new, unlabeled disk.

● **Type `y` to label the disk:**

```
Disk not labeled. Label it now? y
```

Creating and Mounting File Systems

You need to create a file system for a partition when you have:

- Added or replaced a disk drive
- Changed the existing partitioning structure

You must know the logical device name of the disk drive and partition for which you want to create a file system. Refer to Appendix A, “SCSI Addressing” for more information on logical device names.

Note – The disk for which you plan to create a file system must already be formatted and partitioned.

▼ To Create a File System

1. **Become superuser.**
2. **Use the `newfs` command to create a new file system on a partition:**

```
# newfs /dev/rdisk/cntndnsn
```



Caution – Make sure that you have specified the correct logical device name *before* you create a new file system. If you specify the wrong logical device name you will render the data on that partition inaccessible. See “Logical Device Names” on page 61 for details.

3. **Type `y` to confirm:**

```
newfs: construct a new file system /dev/rdisk/cxtxdxsx (y/n)? y
```

`Newfs` creates the necessary file system structures and a `lost+found` directory on the disk partition. Repeat these steps for each partition that will be used to contain a file system. Only one file system can be created per partition.

▼ To Mount a File System

Once you have created a file system, you need to make it available to others by mounting it. A mounted file system is *attached* to the system directory tree at the specified mount point (a directory), and becomes available to the system and users. The root file system is always mounted. Any other file system can be connected (mounted) or disconnected (unmounted) from the root (`/`) file system.

To mount a file system, follow these steps:

1. **Edit the `/etc/vfstab` file with `vi` or any text editor.**

- a. Add the entry by separating each field with a space or a tab.
 If a field has no value, enter a dash (-) as a field placeholder. There are seven fields and each field must have an entry.

```
#device      device      mount      FS   fsck      mount      mount
#to mount    to fsck     point      type pass    at boot    options
#
#/dev/dsk/c1d0s2 /dev/rdisk/c1d0s2 /usr   ufs  1      yes      -
fd         -         /dev/fd fd         -         no       -
/proc      -         /proc  proc      -         no       -
/dev/dsk/c0t0d0s1 -         -       swap      -         no       -
/dev/dsk/c0t0d0s0 /dev/rdisk/c0t0d0s0 /       ufs      1 no     -
swap       -         /tmp   tmpfs     -         yes      -
/dev/dsk/c0t1d0s6 /dev/rdisk/c0t1d0s6 /abc   ufs      2      yes      -
/dev/dsk/c0t1d0s7 /dev/rdisk/c0t1d0s7 /def   ufs      2      yes      -
```



Caution – Be careful when editing this system configuration file. The field columns do not line up. It is very easy to make a mistake. The system reads this file at boot time and when `mount` and `fsck` commands are run. If an error resides in this file, these commands may fail, leaving the system in an undesirable state.

In this example, the bottom two lines were added, one line for each new file system. Each field was entered based on the following field information:

TABLE 2-1 fields /etc/vfstab

Field	Field Heading	Description
1	device to mount	Specify the logical device name for the file system that will be mounted, usually from the <code>/dev/dsk</code> directory.
2	device to fsck	Specify the logical device name of the device that <code>fsck</code> runs on, usually from the <code>/dev/rdsk</code> directory.
3	mount point	Specify the directory to use as the file system mount point.
4	FS type	Specify the file system type, usually <code>ufs</code> for local file systems.

TABLE 2-1 fields (Continued) (Continued)/etc/vfstab

Field	Field Heading	Description
5	fsck pass	Specify a number that controls the way that <code>fsck</code> runs: 1=run <code>fsck</code> one file system at a time, in the order listed. 2=run <code>fsck</code> simultaneously on file systems.--do not run <code>fsck</code>
6	mount at boot	Specify <code>yes</code> or <code>no</code> . <code>yes</code> =this file system should be mounted at boot time and with the <code>mountall</code> command. <code>no</code> = this file system should not be mounted at boot time nor with the <code>mountall</code> command.
7	mount options	Specify mount options separated by a comma but no whitespace, or place a dash (-) in this field for no options.

b. Save the /etc/vfstab file with the modifications.

Note – For more information about writing entries in the `/etc/vfstab` file, see *man Pages(4): File Formats and the System Administration Guide*.

2. Create the directories that you want to use as mount points:

```
# mkdir /abc /def
```

The mount point directory must be created prior to mounting a file system. Two mount points (directories) are created for this example.

3. Type:

```
# mount /abc /def
```

The `mount` command will read the `/etc/vfstab` entries for these file systems and mount them accordingly. Also, each time you boot the system the `/etc/vfstab` file is read and all appropriate file systems, including the new ones, will be mounted.

Note – Be sure not to mount any partition that you intend to use as swap space or as an unmounted partition.

4. Use the `df` command to see all file systems that are mounted:

```
# df -k
```

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/dsk/c0t0d0s0	1759982	779007	975109	45%	/
/proc	0	0	0	0%	/proc
fd	0	0	0	0%	/dev/fd
swap	217808	384	217424	1%	/tmp
/dev/dsk/c0t1d0s6	118344	4	118340	0%	/abc
/dev/dsk/c0t1d0s7	80059	4	80055	0%	/def

● To unmount a file system use the `umount` command:

```
# umount /abc
```

You must be superuser to mount and unmount file systems.

The operating system will not permit you to unmount the `/` or `/usr` file systems.

If your current working directory is in the file system that you wish to unmount, you must change (`cd`) to another directory or the system will report `device is busy`.

Setting Up and Using a Tape Drive

This chapter describes how to configure and use tape drive devices with the Solaris™ operating environment.

This chapter contains the following information:

- “Requirements” on page 27
- “To Add a Tape Drive” on page 29
- “To Display the Status of a Tape Drive” on page 30
- “To Tension a 1/4-inch Tape Cartridge” on page 31
- “To Rewind a Tape” on page 32
- “To Clean a Tape Drive” on page 32
- “Tape Utilities” on page 32

Requirements

Before you start to configure the operating environment for a new tape drive, you must:

- Install the Solaris software
- Shut down the system and peripherals that are already connected, as described in “Shut Down the System” on page 3

Note – If you did not follow the procedures in “Shut Down the System” on page 3, the operating system may not recognize the new tape drive.

Tape Drives

Tape drives are used primarily to store data offline, often as a backup.

Once a new tape drive is connected to the system and booted properly, the tape drive is ready for use. Unlike disk drives, tape drives do not need to be formatted, labeled, or prepared in any special way.

Once the tape device is installed and the operating system is running, you refer to the tape device by the logical device name as described in the TABLE 3-1 below. For more information on logical device names refer to “SCSI Addressing” on page 57.

TABLE 3-1 Address and Logical Device Names for Tape Drives

Tape Drive	SCSI Address Switch or Jumper Setting (Target ID)	Primary Logical Device Name	Additional Logical Device Names
First tape drive	4*	/dev/rmt/0	/dev/rmt/0l /dev/rmt/0m /dev/rmt/0h /dev/rmt/0u
Second tape drive	5*	/dev/rmt/1	/dev/rmt/1l /dev/rmt/1m /dev/rmt/1h /dev/rmt/1u

*Can be any unique address on the bus.

For most tape operations you should use the *primary* logical device name because the tape drive will use its optimum default characteristics. However, if you want to specify a particular tape drive behavior, append a letter to the appropriate logical device name as follows:

- Append a letter to the drive number to specify a *tape density* where h is high, m is medium, l is low, and u is ultra. Not all tape drives support all densities. If you are in doubt, specify the highest density.
- Append a b to the drive number to specify *BSD behavior*. This means that when reading past an end-of-file mark, it returns the first record of the next file. This is desirable if the tape is going to be read on a BSD UNIX system.
- Append an n to the drive number to specify *no rewind* operation, otherwise the tape will rewind when the tape operation is complete.
- Append a c to specify compression mode. For some drives the compression mode is incorporated in some of the data density codes and adding the c is redundant.

Valid tape names are shown in the table below. If you are working with the second tape drive use a 1 instead of 0.

0n	0c	0h	0m	0l	0u
0b	0cb	0hb	0mb	0lb	0ub
0bn	0cbn	0hbn	0mbn	0lbn	0ubn
	0cn	0hbn	0mn	0ln	0un

▼ To Add a Tape Drive

1. Shut down your system.

See “Shut Down the System” on page 3.

2. Check that the address switch for the tape drive has been set correctly.

The SCSI address switch for tape drives is typically preset by the factory to SCSI target ID 4. If you have more than one tape drive, you must set the SCSI address switch of the second tape drive to an address other than SCSI target ID 4 or any other target ID present on the bus.

Refer to the documentation that came with your new tape device and to Appendix A, “SCSI Addressing”.

Note – Connecting more than four tape drives to a single SCSI bus can cause slower system performance.



Caution – Each tape drive must have a unique SCSI target ID setting. The SCSI address switches on your tape drives may need to be set to different numbers than those shown in TABLE 3-1. For further information on device addresses, see the *System Administration Guide*.

3. Install the tape drive.

For installation information, refer to the documentation that accompanies your hardware.

4. Turn the power on to the system and all the peripheral devices.

In most cases, this will cause the system to automatically boot. If the `/reconfigure` file is present (as described in “To Prepare the System” in Chapter 1), then the operating system will automatically assign a logical device name and initialize the

appropriate device drivers for the new device. If the `/reconfigure` file does not exist, you can achieve the same effect by performing a `boot -r`. See Appendix B, “Booting Your System” for more information.

Your new tape drive is ready for use.

Miscellaneous Tape Commands

▼ To Display the Status of a Tape Drive

1. Load a tape into the tape drive.
2. Enter:

```
% mt -f /dev/rmt/unit_number status
```

This command locates the tape drive with the `/dev/rmt/unit_number` logical device name. It then displays the status of the tape drive. (See the examples below.)

If you have more than one tape drive connected to your system use the appropriate logical device name such as `/dev/rmt/0`, `/dev/rmt/1`, or `/dev/rmt/2`, and so on.

- Your system is able to access the tape drive and a tape cartridge is present, if the following status is displayed:

```
Quantum DLT7000 tape drive:
sense key(0x0)= nosense      residual= 0 retries= 0
file no= 0      block no= 0
```

`sense key= nosense` indicates that your system was able to access the tape and there were no errors. You can then use any of the backup commands described in the following sections.

- The following information tells you that the tape drive was just reset, powered on, or a tape cartridge was recently installed.

```
Quantum DLT7000 tape drive:
sense key(0x6)= unit attention      residual= 0 retries= 0
file no= 0      block no=0
```

In this case, execute the `mt -f /dev/rmt/0 status` command again and the `sense key(0x6)= unit attention` advisory message will be replaced with the `sense key(0x0)= nosense` message.

- If the following status is displayed, your system is unable to access the tape.

```
/dev/rmt/0: no tape loaded or drive offline
```

In this case, insert a tape and execute the `mt -f /dev/rmt/0 status` command again.

- If the following status is displayed, your system is unable to communicate with the tape drive.

```
no such file or directory
```

In this case, verify that the tape drive is powered on and is properly connected to the system. If the same message appears try rebooting the system as described in “To Boot After Connecting a Peripheral Device” on page 65.

▼ To Tension a 1/4-inch Tape Cartridge

When you insert a *blank* tape cartridge into a 1/4-inch tape drive, you should perform a tensioning pass. This procedure runs the tape from one end to the other and ensures an even distribution of tension throughout the tape cartridge.

Note – The retensioning command is not supported for Digital Data Storage (DDS) or DLT tape drives.

1. Insert the cartridge tape into the tape drive.

2. Type:

```
% mt -f /dev/rmt/unit_number retension
```

▼ To Rewind a Tape

● Type:

```
% mt -f /dev/rmt/unit number rewind
```

▼ To Clean a Tape Drive

Tape drives must be cleaned periodically. To clean a tape drive:

● Insert a cleaning cartridge into the tape drive.

Consult your hardware documentation for details.

Tape Utilities

There are many utilities that are used to read or write data to a tape drive. This section covers some of the common tape utilities that come with the Solaris operating environment. If you have purchased an additional tape or backup utility please refer to that documentation.

The following commands are covered in this section:

- “cpio Command” on page 33
- “dd Command” on page 34
- “ufsdump Command” on page 34
- “ufsrestore Command” on page 35
- “tar Command” on page 37

Note – The blocking factor is the amount of data sent to the drive for each command. It is best to use the largest factor possible. See the `st` man page for more information.

The blocking factor used for reading must be the same or larger than the blocking factor used for writing the data. Thus it is best to use the largest possible blocking factor.

Use a blocking factor of 126 instead of the default factor of 20 to optimize performance.

Blocking factors are specified as options on the command line of the tape utility command that you choose to use.

cpio Command

The `cpio` command (*copy in/out*) copies a list of files between devices by taking a list of names from standard input and writing to standard output. It is a versatile command but requires you to redirect standard-in and standard-out. `cpio` can be used when the amount of data requires the use of more than one volume (tape cartridge).

Example 1:

The following example shows how to copy the files in your working directory called `/work` and all subdirectories to the first tape drive with the logical device name of `/dev/rmt/0`.

```
example# cd /work
example# ls -R | cpio -ocB > /dev/rmt/0
```

Example 2:

The next example shows how to copy the files that are located on your tape back to your hard disk:

```
example# cd /work
example# cpio -icdB < /dev/rmt/0
```

- The `o` option (*copy out*) reads the standard input to obtain a list of path names and copies those files onto the standard output.

- The `i` option (*copy in*) extracts files from the standard input.
- The `c` option indicates that header information has been written in ASCII format for portability.
- The `d` option indicates that as many directories as needed will be created.
- The `B` option, which you must use whenever you copy files or file systems to and from a tape drive, indicates that the input has a blocking factor of 5120 bytes to the record.

Note – You must use the same blocking factor when you retrieve or copy files from the tape to the hard disk as you did when you copied files from the hard disk to the tape. In such cases you must specify the `B` option.

dd Command

This command converts and copies files with different data formats. One usage of this command is to transfer a file system or partition from your hard disk to a tape. You can also use it to copy files from one hard disk to another. If you use a device with a variable block size, you need to make sure that you use the same block size when you retrieve the data.

The following example shows how to write the file system or partition `/user/sunsystem` to a 4mm tape drive with a tape drive called `/dev/rmt/0`. The blocking factor is 63k in this example.

Example:

```
example# dd if=/dev/dsk/c0t0d0s7 of=/dev/rmt/0 bs=63k
```

ufsdump Command

The `ufsdump` command copies a file system or directory from a hard disk to a tape. It is specifically designed as a backup utility. It supports multiple volumes, and incremental backups. For more information about incremental dump levels and schedules, see the *System Administration Guide*.

Note – All files must be part of one file system or one partition and you must be superuser.

Example:

The following example shows how to copy all files that are located on a disk drive in partition `/dev/rdisk/c0t3d0s0` to a *dump file* (that represents the tape drive in this case) called `/dev/rmt/1`.

```
example# ufsdump 0ubf 126 /dev/rmt/1 /dev/rdisk/c0t1d0s0
```

- The 0 (zero) option represents the *dump level*. A level 0 dump copies the entire file system to a dump file (tape drive). Valid integers are 0 through 9, where 1 through 9 perform various levels of incremental dumps and would only copy files that have changed from a certain time.
- The u option updates the dump record by adding a time-stamped entry to the `/etc/dumpdates` file for each file system that is successfully copied.
- The b option specifies the blocking factor that is used when the files are copied to the tape. The b option must be followed by a number (96 in this case) that represents your desired blocking factor. The default blocking factor varies depending on the type of tape drive that you use. If you don't know what blocking factor to use, do not specify one and the default value for your tape drive will work.
- The f option specifies the *dump file* (logical device name of the tape drive). The f option must be followed by a file name, which is `/dev/rmt/1` in this example.
- `/dev/rdisk/c0t1d0s0` is the name of the source that contains the files that you want to write to tape. In this example, all the files will be copied from the first partition (s0) of the second disk drive (t1).

ufsrestore Command

The `ufsrestore` command copies file systems from a tape to a hard disk. It can only copy file systems that were created with the `ufsdump` command.

You must use `ufsrestore` with one of these three options:

- i (interactive) – After reading the directory information from the media, `ufsrestore` invokes an interactive interface that allows you to browse through the dump file's directory hierarchy and select individual files to be extracted.
- r (recursive) – Restore the entire contents of the media into the current directory (which should be the top level of the file system). To completely restore a file system, use this option to restore the level 0 dump, and again for each incremental dump. Although, this option is intended for a complete restore onto a clear file system, if the file system contains files not on the media, they are preserved.
- t (table of contents) – Provides a list of every file on a tape.

The following options are not mandatory:

- **b** (blocking factor) – Specify the blocking factor for tape reads. This option must be followed by an integer. By default, the `ufsrestore` command attempts to determine the correct blocking factor to use.
- **f** (dump file) – Specify the dump file to restore from. `ufsrestore` will use `/dev/rmt/0` if no dump file is specified.
- **v** (verbose mode) – Display path names as they are read from the tape.

Example:

The following example shows how to retrieve a file from tape interactively. The file will be written in the current directory (`/disk2` in this example). A blocking factor of 96 is specified as well as the dump file (first tape drive) of `/dev/rmt/0`.

Note – You must use the same blocking factor (or larger) when you retrieve or copy files from the tape to the hard disk as you did when you copied files from the hard disk to the tape.

```
example# cd /disk2
example# ufsrestore ibf 126 /dev/rmt/0
ufsrestore > ?
Available commands are:
  ls [arg] - list directory
  cd arg - change directory
  pwd - print current directory
  add [arg] - add 'arg' to list of files to be extracted
  delete [arg] - delete 'arg' from list of files to be extracted
  extract - extract requested files
  setmodes - set modes of requested directories
  quit - immediately exit program
  what - list dump header information
  verbose - toggle verbose flag (useful with "ls")
  help or '?' - print this list
IF no 'arg' is supplied, the current directory is used
```

The system responds with a `ufsrestore` prompt. If you type a question mark, a list of available arguments is displayed.

List the directories that are resident on the tape by typing `ls`.

```
ufsrestore > ls
4lib/      dict      mail      openwin   spool
5bin       games     man/      preserve  src
adm        include/  net       pub       tmp
```

Now add the files or directories to the list of files to extract with the `add` command:

```
ufsrestore > add man mail
```

In this example the `man` and `mail` files are added to the list of files to extract from tape.

Now extract these files from the tape. They will be written to your current working directory (`/disk2` in this example).

```
ufsrestore > extract
```

Once all the desired files are extracted, exit the `ufsrestore` command with `q`:

```
ufsrestore > q
```

tar Command

The `tar` command archives and retrieves files to and from a single file called a *tarfile*. A tarfile is usually a tape, but it can be any file. If the amount of data exceeds one tape then you need to use a different tape command because `tar` cannot span multiple volumes (tapes).

Example 1:

The following example shows how to copy files from a hard disk to a tape.

```
example# tar cvbf 126 /dev/rmt/1 ./directory
```

In this example, the `tar` command copies files from a *directory* to a tape drive with the device name of `/dev/rmt/1`. Specifying the directory source preceded with a `./` will cause the files to be recorded as relative path names. Later, when the files are extracted, they files will be written back into the file system based on your current working directory.

- The `c` option *creates* the tarfile (writes to tape in this case).
- The `v` option displays information about each file it copies (verbose mode).
- The `b` option designates the blocking factor, which in this example is `96`.
- The `f` option designates the device name of the source drive, which is the tape drive in this example.

Example 2:

The next example shows how to copy files from a tape to the current working directory on a hard disk. It is very important to change to the directory where the extracted files will reside *before* running `tar`.

```
example# cd /tmp
example# tar xvbf 126 /dev/rmt/1
```

- The `x` option *extracts* files from the tarfile (tape).
- The `v` option displays information about each file it is extracted (*verbose* mode).
- The `b` option designates the *blocking factor*, which in this example is 126 (63k).
- The `f` option allows you to designate the *dump file* of the destination drive.

Note – You must use the same blocking factor (or larger) when you retrieve or copy files from the tape to the hard disk as you did when you copied files from the hard disk to the tape.

Setting Up Removable Media Devices

This chapter describes how to configure removable media devices using the Solaris operating environment. Devices commonly referred to as removable media devices include:

- CD-ROM drives
- Diskette drives
- PCMCIA memory cards

This chapter contains the following information:

- “Requirements” on page 39
- “To Add a CD-ROM Drive” on page 40
- “To Add an Additional CD-ROM Drive” on page 41
- “To Add a Diskette Drive” on page 42
- “To Format a PCMCIA Memory Card” on page 43
- “How Solaris Manages CD-ROM and Diskette Devices” on page 44

Requirements

Before you start to configure the operating environment for a new CD-ROM, diskette drive, or PCMCIA memory card, you must:

- Install the Solaris software
- Shut down the system and peripherals that are already connected, as described in “Shut Down the System” on page 3

Note – If you did not follow the procedures in “Shut Down the System” on page 3, the operating system may not recognize the new removable media device.

CD-ROM Drives

CD-ROMs provide large data capacity and have quickly become today’s medium of choice for software distribution. CD-ROM drives allow you to add both operating system and application packages quickly and easily to the workstations you administer.

Most CD-ROM drives conform to the Small Computer Systems Interface (SCSI) protocol. As a result, getting a CD-ROM drive up and running on your workstation usually involves little more than plugging in the cables and, if applicable, setting the jumpers or SCSI target switch to 6 (or another number if it is a secondary drive). Refer to the documentation that accompanies your hardware for information about physical connections.

Note – For more information about CD-ROMs, refer to “Managing CDs and Diskettes” in the *System Administration Guide*.

▼ To Add a CD-ROM Drive

1. Shut down your system.

See “Shut Down the System” on page 3.

2. Check that the address switch for the CD-ROM drive has been correctly set.

The SCSI address switch for CD-ROM drives is usually preset by the factory to SCSI target ID 6.

Note – If you have more than one SCSI device, you must set each SCSI address switch to a different target ID.

Note – Refer to the documentation that accompanies your hardware and “SCSI Addressing” on page 57 to set the physical SCSI switch or jumpers on your CD-ROM drive.

3. Install the CD-ROM drive.

For installation information, refer to the documentation that accompanies your hardware.

4. Turn on the power to all external peripheral devices first. This includes the CD-ROM drive that you plan to configure.

5. When all external peripheral devices have been powered on, power on the system and use the `boot -r` command at the boot prom level.

```
ok boot -r
```

The system is booted and the Solaris operating environment configures the system by assigning a device address to the CD-ROM drive.

You are now ready to mount the CD-ROM drive. Refer to “How Solaris Manages CD-ROM and Diskette Devices” on page 44.

▼ To Add an Additional CD-ROM Drive

Note – Be sure that the target number you select is not already used by a peripheral device attached to your system. Use the `probe-scsi` or `probe-scsi-all` command to list the existing devices and their addresses. You can connect no more than seven CD-ROM drives to a single SCSI bus provided that the total SCSI bus length does not exceed 6 meters.

1. Shut down your system.

See “Shut Down the System” on page 3.

2. Install the additional CD-ROM drive.

For installation information, refer to the documentation that accompanies your hardware.

3. Turn on the power to all external peripheral devices, this includes the second CD-ROM drive that you plan to configure and the system.

The system boots and the Solaris operating environment configures the system by assigning a device name to the CD-ROM drive.

You are now ready to mount the CD-ROM drive. Refer to “How Solaris Manages CD-ROM and Diskette Devices” on page 44.

Diskette Drives

Diskette drives provide small amounts of data storage and are often used to transfer files to a non-networked personal computer or to store individual user files.

The diskette drives that Sun uses are not SCSI devices. They are connected to a diskette controller that is built into the CPU board.

▼ To Add a Diskette Drive

1. **Shut down your system.**

Note – See “Shut Down the System” on page 3.

2. **Install the diskette drive.**

For installation information, refer to the documentation that accompanies your hardware.

3. **Turn on the power to the system and use the `boot -r` command at the boot prom level.**

```
ok boot -r
```

The system automatically configures the new diskette drive.

You are now ready to mount the diskette drive. Refer to “How Solaris Manages CD-ROM and Diskette Devices” on page 44.

PCMCIA Memory Cards

Personal Computer Memory Card International Association (PCMCIA) cards are rugged, credit card-sized, user-installable devices. You can use PCMCIA memory cards in the same way as a diskette, but you can store much larger amounts of data on a PCMCIA memory card.

PCMCIA serial and modem cards provide a convenient way to add an RS-232 interface or data/fax modem functionality to your SPARCstation. PCMCIA cards are available from many vendors. Check with your SunService provider or the PCMCIA card vendor to determine if a device is compatible with your SPARCstation.

▼ To Format a PCMCIA Memory Card

Before using a PCMCIA memory card, you may need to format it. Use the `fdformat` utility to format diskettes and PCMCIA memory cards.



Caution – Formatting deletes all data on a PCMCIA memory card. (See the “`fdformat` Utility Options table below.)

- **Format the PCMCIA memory card using the `fdformat` command:**

```
% fdformat option device_name
```

The following table lists the available options for the `fdformat` utility.

TABLE 4-1 `fdformat` Utility Options

Option	Description
<code>-U</code>	Unmounts the PCMCIA memory card
<code>-d</code>	Installs an MS-DOS file system (UNIX file system is the default)
<code>-f</code>	Does not display confirmation messages before starting to format
<code>-q</code>	Disables print status messages
<code>-x</code>	Installs a Solaris label or an MS-DOS file system; it does not format the PCMCIA memory card
<code>-b label</code>	Specifies a UNIX or MS-DOS label on a PCMCIA memory card
<code>-t dos</code>	Installs an MS-DOS file system (UNIX file system is the default)
<code>-B filename</code>	Installs a special boot loader

Note – There is no option in the `fdformat` utility for installing a NEC-DOS file system on a PCMCIA memory card.

If you want to format a PCMCIA memory card, you must specify a device name for the PCMCIA memory card. Otherwise, the `fdformat` utility automatically specifies the diskette drive as the default device.

The format for a device name of a PCMCIA memory card is `/dev/rdsk/cntndnsn` or `/dev/dsk/cntndnsn`. For example, the device name `/dev/dsk/clt6d0s2` represents a PCMCIA SRAM memory card with the logical socket controller number 1, the technology number 6, and the slice number 2.

TABLE 4-2 lists the available device name options for the `fdformat` utility.

TABLE 4-2 `fdformat` Utility Device Name Options

Device Name Option	Description
<i>n</i>	A decimal number
<i>cn</i>	Controller <i>n</i>
<i>tn</i>	Technology region <i>n</i> 0x1 ROM, 0x2 OTPROM, 0x3 EPROM, 0x4 EEPROM, 0x5 FLASH, 0x6 SRAM, 0x7 DRAM
<i>dn</i>	Technology region in type <i>n</i>
<i>sn</i>	Slice <i>n</i>

Note – This release supports only one partition on the PCMCIA memory card. Therefore, the partition number *sn* for the device name must be *s2*.

How Solaris Manages CD-ROM and Diskette Devices

This section describes how CD-ROM and diskette devices are managed by the Solaris operating environment.

Volume Management

Volume Management is a program in the operating system that automates the administration of your CD-ROMs and diskettes. For example, in previous Solaris releases (Solaris 2.1 or earlier), to mount and access data on a CD-ROM you had to perform many administrative steps (refer to “*Comparing Automatic and Manual Mounting*” on page 47).

In the Solaris operating environment, Volume Management automatically mounts CD-ROMs and diskettes with file systems at `/cdrom/cdrom_name` and `/floppy/floppy_name` respectively. It also keeps track of CD-ROM and diskette file systems during a workstation session (rebooting will clear the in-memory database). To view the media that has been inserted during a workstation session, list `/vol/dsk`:

```
% ls /vol/dsk
supp_sol_2_7_smcc
unnamed_cdrom
unnamed_floppy
```

Volume Management uses the configuration file `/etc/vold.conf` to determine which devices it manages. The default `/etc/vold.conf` file contains the following information:

```
# @(#)vold.conf 2.21    96/05/10 SMI
#
# Volume Daemon Configuration file
#
# Database to use (must be first)
db db_mem.so

# Labels supported
label dos label_dos.so floppy
label cdrom label_cdrom.so cdrom
label sun label_sun.so floppy

# Devices to use
use cdrom drive /dev/rdisk/c*s2 dev_cdrom.so cdrom%d
use floppy drive /dev/rdiskette[0-9] dev_floppy.so floppy%d
use pcmem drive /dev/rdisk/c*s2 dev_pcmem.so pcmem%d
forceload=true

# Actions
insert /vol*/dev/fd[0-9]/* user=root /usr/sbin/rmm
insert /vol*/dev/dsk/* user=root /usr/sbin/rmm
eject /vol*/dev/fd[0-9]/* user=root /usr/sbin/rmm
eject /vol*/dev/dsk/* user=root /usr/sbin/rmm
notify /vol*/rdsk/* group=tty /usr/lib/vold/volmissing -c

# List of file system types unsafe to eject
unsafe ufs hfs pcfs
```

Before you add secondary CD-ROM and diskette drives to a system, update the `/etc/vold.conf` file by adding the new devices to the “Devices to use” list. The syntax for a “Devices to use” entry is:

```
use device type special shared_object symname options
```


Each variable item in the device control line is defined in TABLE 4-3.

TABLE 4-3 Device Control Syntax Descriptions

Syntax	Description	Supported and Default Values
<i>device</i>	The type of removable media device to be used	cdrom, diskette
<i>type</i>	The class of device: multiple or single media support	drive
<i>special</i>	Path name of the device to be used. Path usually begins with /dev	Default support is for the devices /dev/dsk/c0t6 and /dev/rdiskette
<i>shared_object</i>	The location of the code that manages this device	The default location is /usr/lib/vold/ <i>name_of_shared_object</i>
<i>symname</i>	The symbolic name that refers to this device; the <i>symname</i> is placed in the device directory, either /cdrom or /floppy	Default values are cdrom0, floppy0
<i>options</i>	The user, group, and mode permissions for the media inserted	Default values are user=nobody, group=nobody, mode=0666

After updating the `/etc/vold.conf` file, attach the drive and reboot your system with a reconfiguration boot.

Note – For information on booting your system in different modes, see “To Boot After Connecting a Peripheral Device” on page 65.”

Note – Volume Management controls the `/dev/dsk/c0t6d0s2` path to a CD-ROM drive and the `/dev/diskette` path to a diskette drive. An attempt to access a CD-ROM or diskette using these paths will result in an error message.

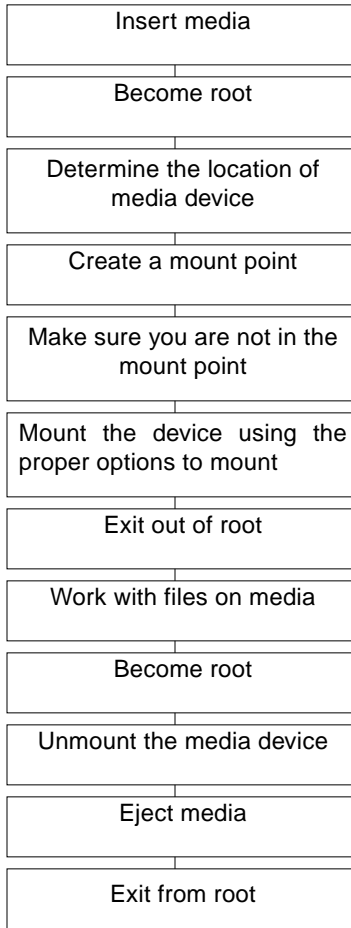
Note – For more information about Volume Management, see *man Pages(4): File Formats* and the *System Administration Guide*.

Comparing Automatic and Manual Mounting

FIGURE 4-1 compares the steps for manual mounting (without Volume Management) and automatic mounting (with Volume Management). Volume Management performs many of the tasks previously required to manually mount and work with CDs and diskettes.

Manual Mounting

Tasks required to work with media



Automatic Mounting

Tasks required to work with media

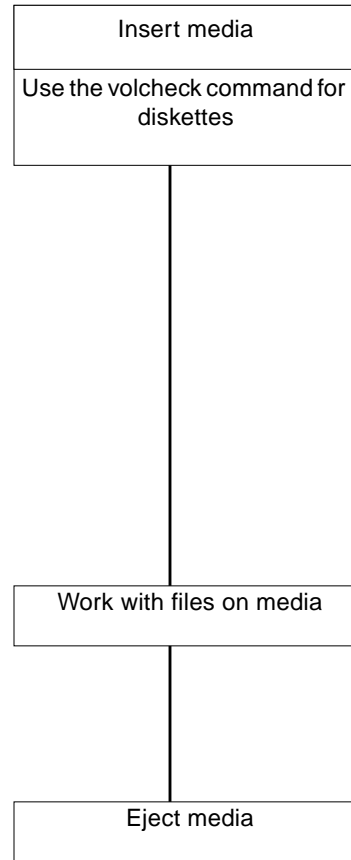


FIGURE 4-1 Comparison of Manual and Automatic Mounting

Reference Material for CD-ROM and Diskette Devices

CD-ROM and diskette file systems are mounted in default locations by Volume Management when the media is inserted. See TABLE 4-4 for more information. These mount points are only created and mounted if a file system is already resident on the diskette.

TABLE 4-4 CD-ROM and Diskette File System Mount Points

Media type	Mount location	State of media
Diskette	<code>/floppy/floppy0</code>	Symbolic link to mounted diskette in local diskette drive
	<code>/floppy/floppy_name</code>	Mounted named diskette
	<code>/floppy/unnamed_floppy</code>	Mounted unnamed diskette
CD-ROM	<code>/cdrom/cdrom0</code>	Symbolic link to mounted CD-ROM in local CD-ROM drive
	<code>/cdrom/CD-ROM_name</code>	Mounted named CD-ROM
	<code>/cdrom/CD-ROM_name/partition</code>	Mounted named CD-ROM with partitioned file system
	<code>/cdrom/unnamed_cdrom</code>	Mounted unnamed CD-ROM

To inform Volume Management that a diskette with a file system has been inserted into the drive, use `volcheck(1)`. You do not need to use this command if you are going to use `fdformat(1)` or `eject(1)`.

Note – For more information about these commands, see `man Pages(1): User Commands` and the *System Administration Guide*.

If no file system exists on the media, Volume Management provides block and character devices in the `/vol` file system. See TABLE 4-5 for the location of diskette and CD-ROM media in the `/vol` file system.

Note – All releases provide a symbolic link in `/vol/dev/aliases`.

TABLE 4-5 CD-ROM and Diskette Device Locations in `/vol` With No File System Present

Media Type	Device Location:	State of Media
Diskette	<code>/vol/dev/diskette0/unnamed_floppy</code>	Formatted unnamed diskette—block device access
	<code>/vol/dev/rdiskette0/unnamed_floppy</code>	Formatted unnamed diskette—raw device access
	<code>/vol/dev/diskette0/unlabeled</code>	Unlabeled diskette—block device access
	<code>/vol/dev/rdiskette0/unlabeled</code>	Unlabeled diskette—raw device access
CD-ROM	<code>/vol/dev/dsk/c0t6d0/unnamed_cdrom</code>	CD-ROM—block device access
	<code>/vol/dev/rdsk/c0t6d0/unnamed_cdrom</code>	CD-ROM—raw device access

Using CDs and Diskettes

TABLE 4-6 shows the various tasks you can perform using CDs and diskettes. See the *System Administration Guide*, for details.

TABLE 4-6 What You Can Do With CDs and Diskettes

Media Type	Task	Available with File Manager?	Available through Command Line?
CD	Load a CD	Yes	Yes
	Examine the contents of a CD	Yes	Yes
	Copy information from a CD	Yes	Yes
	Find out if a CD is still in use	No	Yes
	Eject a CD	Yes	Yes
	Access CDs on other systems	No	Yes
	Make local CDs available to other systems	No	Yes
	Configure a system to play musical CDs	No	Yes
	Stop Volume Management	No	Yes
Restart Volume Management	No	Yes	
Diskette	Format a UFS diskette	Yes	Yes
	Place a UFS file system on a diskette	Yes	Yes
	Format a DOS diskette	Yes	Yes
	Load a diskette	Yes	Yes
	Examine the contents of a diskette	Yes	Yes
	Copy or move information from a diskette	Yes	Yes
	Copy or move information to a diskette	Yes	Yes
	Find out if a diskette is still in use	No	Yes
	Eject a diskette	Yes	Yes
	Access diskettes on other systems	No	Yes
Make local diskettes available to other systems	No	Yes	

Setting Up an Option Card

This chapter describes how to configure a option card using the Solaris™ software environment.

This chapter contains the following information:

- “Option Cards” on page 53
- “Requirements” on page 54
- “To Add an Option Card” on page 54

Option Cards

Many different kinds of cards are supported in Sun systems. Option cards provide the means to add options to your system with the ease of *plug and play*.

The kinds of options that you can add to a system are too numerous to list here, but some of the common options include:

- Peripheral interfaces such as SCSI, IDE, and Fibre Channel Host Adapters
- Frame Buffers for video and graphics support
- Communication and network interfaces
- Bus extension cards

The *type* of option card that you add to your system depends on the bus in your system. The systems sold today have SBus or PCI busses and require a corresponding option card. Also some frame buffer option cards connect directly to the UPA bus.

Requirements

Before you start to configure the operating environment for a new card, you must:

- Install the Solaris software
- Shut down the system and peripherals that are already connected, as described in “Shut Down the System” on page 3

Note – If you did not follow the procedures in “Shut Down the System” on page 3, the operating system may not recognize the new card.

To Add an Option Card

1. Shut down your system.

See “Shut Down the System” on page 3

2. Determine the address selection scheme of card.

In most cases, the card becomes addressed when you plug it in. The address will be based on the connector that it is plugged in to.

Note – For more information, refer to the documentation that accompanies your hardware.

3. Set any jumpers or switches that require different settings.

For example, if your system has an SBUS and if you are adding an Ethernet card, you may need to change one jumper on that card related to the Link Integrity Test.

4. Install the card.

For installation information, refer to the documentation that accompanies your hardware.

5. Turn the power on to the system and all the peripheral devices.

In most cases, this will cause the system to boot. If the `/reconfigure` file is present (as described in “To Prepare the System” in Chapter 1), then the operating system will automatically initialize the appropriate device drivers for the new card. If the

`/reconfigure` file does not exist, you can achieve the same effect by performing a `boot -r`. See “Booting Your System” on page 65” in the appendix for more information.

6. If you want to display a list of all the devices on your system, use the `prtconf` command as shown:

```
example# prtconf | grep -v not
System Configuration: Sun Microsystems sun4u
Memory size: 128 Megabytes
System Peripherals (Software Nodes):
SUNW,Ultra-5_10
  options, instance #0
  pci, instance #0
    pci, instance #0
      ebus, instance #0
        power, instance #0
        se, instance #0
        su, instance #0
        su, instance #1
        fdthree, instance #0
      network, instance #0
      ide, instance #0
        dad, instance #0
        sd, instance #2
    pci, instance #1
      SUNW,ffb, instance #0 (card installed here)
  pseudo, instance #0
```


SCSI Addressing

Achieving the appropriate peripheral address depends on the *type* of peripheral subsystem that you have. Sun systems support a variety of peripheral subsystems such as:

- SCSI – Small Computer System Interface (covered in this appendix)
- IDE – Integrated Drive Electronics
- SOC – Serial optical controller
- IPI – Intelligent Peripheral Interface

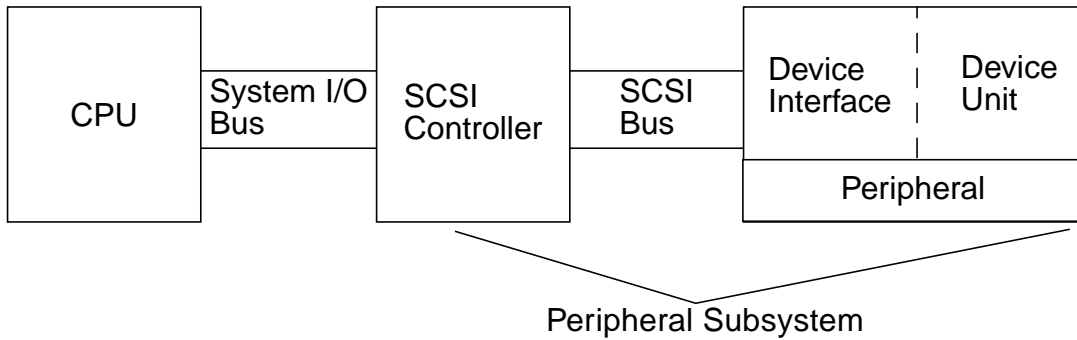
Refer to the documentation that accompanies your hardware to determine the type of peripheral subsystem that you have.

This chapter describes addressing concepts for the SCSI peripheral subsystem.

SCSI Subsystem

To understand how to address and access a SCSI device you need to understand how the device is connected to the system. The following section describes how SCSI systems are connected to a system.

The following block diagram represents a typical SCSI subsystem and how it is connected to the system.



- CPU – is the central processing unit of the system.
- System I/O Bus – is inherent to the design of the system. You might have a system with a PCI, SBus, VME, or combination of these.
- SCSI controller – interprets the electrical signals between the system I/O bus and the SCSI bus. For more information refer to “SCSI Controller” below.
- SCSI Bus – is the path from the SCSI controller to the Device interface.
- Device Interface – interprets the electrical signals between the SCSI bus and the device. It is usually built in to the device unit.
- Device Unit – is the actual peripheral such as a disk, tape, or CD-ROM device. A Device unit with a built-in interface makes up the whole peripheral.
- Device Driver – (not shown) is software that enables the operating system to communicate with the device.

The combination of SCSI controller, SCSI bus, device interface and device unit make up the *peripheral subsystem*. All of these entities must be designed and configured to work together.

SCSI Controller

A SCSI controller (sometimes referred to as *SCSI host*) is provided in two ways; as circuitry built-in to the main logic board (CPU board) often referred to as an “onboard” interface; or added to a system by way of a card plugged into the system I/O bus.

If you are adding a SCSI controller card to a system you must add a card that corresponds to your type of *system I/O* bus.

Typically, your sales representative provides you with the right interface card to fit the system I/O bus and the type of interface that the peripheral device requires. Many systems have built-in SCSI support, and you would not need to install any additional cards.

If you do need to install a SCSI controller card, and it is either an SBus card or a PCI card, the card will be logically addressed automatically based on the order and connector that it is plugged in to. See “Setting Up an Option Card” on page 53” for more information.

SCSI Address Selection Schemes

To configure disk, tape, and CD-ROM drives you need to understand the *address selection scheme* that your system uses. Address selection schemes for disk drives differ from address selection schemes for tape drives. This section discusses the SCSI address selection schemes for different types of peripheral devices.

Addresses and Device Names

The Solaris software identifies a peripheral through a series of addresses and device names:

- Target ID address – an address that is set on the interface of the device by you, or the installer.
- Physical device name – is assigned by the system firmware based on its physical connection to the system.
- Logical device name – is assigned by the operating system when the peripheral is installed.

Target ID Addressing Methods

The target ID is an address that is set on the interface of the device (disk, tape, or CD-ROM). There are three methods to set this address:

- Jumpers – an address is achieved by placing jumpers on shunts. This is how internal CD-ROM devices are addressed. If your device requires jumper installation consult the documentation that shipped with your device.
- Switch – used when a device is in an external enclosure (an enclosure other than the system chassis), the enclosure provides a switch that you set to your desired target ID address.

- Single-connector – A device with a single connector receives data, power, and address information on a single connector. A device of this sort will automatically become addressed when you install it. This is the most common method of addressing Sun disk drives today.

Determining the correct target ID for your device depends on several conditions:

- The target addressing method (described above) used by your device.
- Existing addresses on the interface – All devices must have a unique target ID.
- Device type – some device types such as CD-ROM and tape drives are most easily identified by the operating system when they are with certain addresses. See the table below:

TABLE A-1 SCSI Target Addresses

Device Order	CD-ROM Targets	Tape Target	Disk Target (Ultra & Enterprise Systems)	Disk Target (Pre-Ultra systems)
1st	6	4	0	3
2nd		5	1	1
3rd			2	2
4th			3	0
5th			4	4
6th			5	5
7th			6	6

Note – Some SCSI controllers support 7 devices, others support up to 15 devices (although you should take into account the effect on I/O). If 15 devices are connected, valid disk target IDs are 0-14.

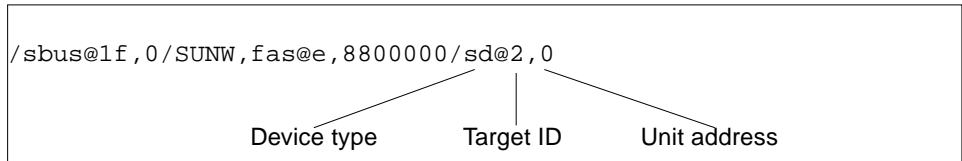
Note – You can not have two devices with the same Target ID on the same SCSI controller. However, target IDs can be reused across different SCSI controllers.

Physical Device Names

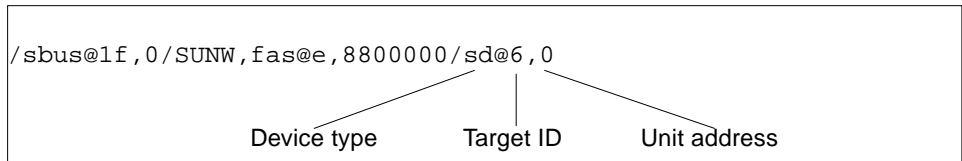
The physical device name is assigned by the system firmware. This name is expressed in the form of a path name. The path name describes the location of the device in relation to the CPU. For SCSI devices the target ID is part of this address name.

After the firmware assigns the physical device names, special files are copied to the `/devices` directory that reflect the physical device names. This occurs when the system is booted with the reconfiguration option (`boot -r` or when `/reconfigure` file exists and the system is booted).

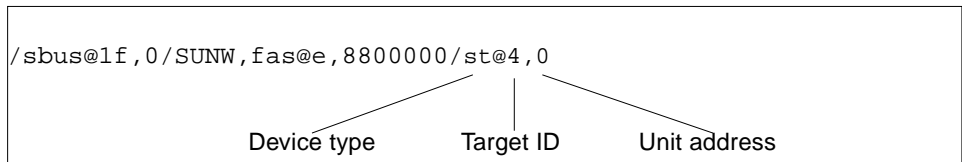
Disk example:



CD-ROM example:



Tape example:



Logical Device Names

The logical device name is created by the operating system when the peripheral is first installed and booted with the reconfiguration option. Logical device names are located in the `/dev` directory. A logical device name is a file that is symbolically linked to the physical device name (names in the `/devices` directory). The file name reflects the address and physical connection of the device to the system. The logical device name is the address that you use when you work with the device.

Logical device names for a disk drive are created as follows:

```
/dev/[r]dsk/cntndnsn
```

- *cn* – is the controller (or interface) number such as *c0*, *c1*, *c2*, and so on. Controller numbers are logically assigned in sequential order. *c0* often represents a built-in SCSI controller.
- *tn* – is the target ID of the device such as *t0*, *t1*, *t2*, *t3* and so on.
- *dn* – is the device number (often called a LUN). It reflects the actual address of the device unit. This is usually *d0* for most SCSI disks because there is only one disk attached to the target controller (device interface).
- *sn* – is the slice number that represents a partition, or *slice*, of a disk. Valid numbers are 0 through 7.

Note – Do not apply the above explanation to disks that are part of a SPARCstorage Array (a device connected to a SOC card). Disks in a SPARCstorage Array have logical device names with the same `/dev/[r]dsk/cntndnsn`; however, they are interpreted differently.

Logical device names for disk drives are created in two subdirectories in the `/dev` directory; `rdsk` and `dsk`. As you use disk logical device names with various commands you must use the appropriate one from either `/dev/rdsk` or `/dev/dsk` depending on whether the command uses a raw (or character) device interface, or a block device interface. The distinction is made based on how data is read from the device:

- **Raw device interface** – A *raw* device transfers data character by character in just the exact amount of data needed for a given task. Use the logical device name from the `/dev/rdsk` directory.
- **Block device interface** – A *block* device transfers data in a predetermined amount, often from a buffer from which large blocks of data are read at once. Use the logical device name from the `/dev/dsk` directory

In general, there are three commands that require a block device: `mount`, `umount`, and `df`. Most other commands function best when you use the raw device. When you're not sure whether a command requires use of `/dev/dsk` or `/dev/rdsk`, check the man page for that command.

CD-ROM logical device names are created following the same scheme as disk drives (see above). The following is an example of a CD-ROM logical device name:

```
/dev/rdsk/cnt6dnsn
```


SCSI tape logical device name example:

```
/dev/rmt/0
```

TABLE A-2 Tape Drive Logical Device Names

Tape Drive	Target ID	Primary Logical Device Name	Additional Logical Device Names
First tape drive	4	/dev/rmt/0	/dev/rmt/0l /dev/rmt/0m /dev/rmt/0h /dev/rmt/0u
Second tape drive	5	/dev/rmt/1	/dev/rmt/1l /dev/rmt/1m /dev/rmt/1h /dev/rmt/1u

For most tape operations you should use the *primary* logical device name because the tape drive will use its optimum default characteristics. However, if you desire to specify a particular tape drive behavior, append a letter (up to three letters) to the appropriate logical device name as follows:

- Append a letter to the drive number to specify a *tape density* where h is high, m is medium, l is low, and u is ultra. Not all tape drives support all densities. If you are in doubt, specify the highest density.
- Append a b to the drive number to specify *BSD behavior*. This means that when reading past an end-of-file mark, it returns the first record of the next file. This is desirable if the tape is going to be read on a BSD UNIX system.
- Append an n to the drive number to specify *no rewind* operation, otherwise the tape will rewind when the tape operation is complete.
- Append a c to specify compression mode. For some drives the compression mode is incorporated in some of the data density codes and adding the c is redundant.

Valid tape name combinations are shown in the table below. If you are working with the second tape drive use a 1 instead of 0.

0l	0c	0h	0m	0l	0u
0b	0cb	0hb	0mb	0lb	0ub
0bn	0cbn	0hbn	0mbn	0lbn	0ubn
	0cn	0hbn	0mn	0ln	0un

Booting Your System

This appendix describes how to boot your system in different modes.

If a system is powered off, turning it on starts the multiuser boot sequence. See below for procedures on how to boot to different states from the `ok` prompt.

- “To Switch to the `ok` Prompt” on page 65
- “To Boot After Connecting a Peripheral Device” on page 65
- “To Boot Automatically” on page 66
- “To Boot to Multiuser State (Init State 3)” on page 67
- “To Boot to Single-User State (Run Level S)” on page 67
- “To Boot Interactively” on page 67
- “To Stop the Boot Process” on page 68

▼ To Switch to the `ok` Prompt

When the system is halted, the PROM monitor prompt is represented by `ok`.

- **Here is an example of a system that has been halted and is at the boot prom level:**

```
ok
```

▼ To Boot After Connecting a Peripheral Device

Booting your system to recognize all newly installed peripheral devices is usually done after:

- Shutting down the operating system
- Powering on the system

- Interrupting the boot process
- When the open boot process is not set for automatic boot
- **Boot your system to recognize all newly installed peripheral devices by typing the `boot -r` command:**

```
ok boot -r
```

Note – The `boot -r` command adds newly installed peripheral devices if you did not perform the `touch /reconfigure` command discussed in “Shut Down the System” on page 3.

▼ To Boot Automatically

Booting automatically is as easy as powering on the system. The system automatically loads the operating system, recognizes all connected devices, and boots to multiuser mode.

Note – To boot automatically, the open boot PROM must be set for automatic boot.

- **To set the Open Boot PROM so your system will automatically boot type the following at the `ok` prompt:**

```
ok setenv auto-boot? true
ok reset
```

To prevent the system from booting when the system is powered on set the `auto-boot` parameter to `false`.

Note – In instances where new hardware has just been added and you did not perform the `touch /reconfigure` command, your system may not automatically configure the new devices. Perform a `boot -r` command at the boot prom level (`ok` prompt). See “Shut Down the System” on page 3 for more information.

▼ To Boot to Multiuser State (Init State 3)

Booting to multiuser state is usually done after halting the system or performing some system hardware maintenance task. This is the default boot level where all resources are available and users can log into the system.

- **Boot to run level 3 by typing the `boot` command:**

```
ok boot
```

The automatic boot procedure starts, displaying a series of startup messages. The system is brought up in multiuser state.

▼ To Boot to Single-User State (Run Level S)

Booting to single-user state is usually done to perform some system maintenance task such as backing up the system. At this level only some file systems are mounted and users cannot log into the system.

1. **Boot to run level S by using the `boot -s` command:**

```
ok boot -s
```

The system boots to single-user state and prompts you for the root password:

```
INIT: SINGLE USER MODE
Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxxxxxxx
```

2. **Enter the root password.**
3. **To bring the system up to multiuser state after the system maintenance task is performed, press Control-D.**

▼ To Boot Interactively

You can boot interactively to make a temporary change to the system file or the kernel. Booting interactively lets you test your changes and recover easily if you have any problems.

This procedure assumes that the system is already shut down.

1. Boot the system interactively, by typing:

```
ok boot -a
```

2. Answer the system prompts as described in the table below.

TABLE B-1 Interactive Boot Procedure Steps

If the System Displays ...	Do the Following ...
Enter filename [kernel/unix]:	Provide the name of another kernel to use for booting. Or, press Return to use the default kernel (kernel/unix).
Name of system file [/etc/system]:	Provide the name of an alternate system file and press Return. Or, press Return to use the default /etc/system file.
Name of default directory for modules [/kernel /usr/kernel]:	Provide an alternate path for the modules directory and press Return. Or, press Return to use the default modules directory path.
root filesystem type [ufs]:	Press Return to use the default root file system type: UFS for local disk booting or NFS for diskless clients.
Enter physical name of root device [/sbus@1,f8000000/esp@0,800000/ sd@3,0:a]:	Provide an alternate device name and press Return. Or, press Return to use the default physical name of the root device.

▼ To Stop the Boot Process

Occasionally, you may need to stop the boot process. The specific key sequence depends on your keyboard type. See Step 1 below.

1. Type the key sequence for your system.

Use one of the following combination of key sequences:

- Stop-A (type-5 keyboards)
- L1-A (type-4 keyboards)

- Break key (TTY terminals only)

The monitor mode command prompt is then displayed on the screen:

```
ok
```

2. To synchronize the disks, type:

```
ok sync
```

3. When you see the `syncing file systems. . .done` message, press the abort key sequence for your system again (Step 1).
4. Type the appropriate `boot` command to restart the boot process.

Index

SYMBOLS

/etc/vold.conf
 default, 46
 updating, 46

A

aborting
 Break, 69
 L1-a, 68
 Stop-a, 68
address
 selecting, 59
 board, 54
 CD-ROM drive, 29, 40

B

backup commands, 32
board
 address selection, 54
booting
 abort, 68
 how to abort
 SPARC, 68
 to multiuser state
 SPARC, 67
 to single-user state
 SPARC, 67
booting to multiuser state
 SPARC, 67

booting to single-user state
 SPARC, 67
Break key, 68

C

CD-ROM drive
 mount points, 49
command
 cpio, 33
 dd, 34
 tar, 37
 ufsdump, 34
 ufsrestore, 35
configuring the system
 board, 54
cpio command, 33

D

dd command, 34
default
 modules directory path
 x86, 68
 root file system type
 x86, 68
 system file (/etc/system)
 x86, 68
device address
 board, 54
device driver, 8

F

file system
 default root
 x86, 68

H

halting a system, 4
 with halt, 4
 with init, 4
 with reboot, 4
 with shutdown, 4

I

init state, 5
interactive
 booting
 SPARC, 67

L

L1-A command, 68

M

modules directory path, default
 x86, 68
mounting a CD-ROM, 41, 42
multiuser state
 booting to
 SPARC, 67

P

peripheral bus
 Sbus, 59
 VMEbus, 59
probe-scsi command, 7, 10
PROM
 prompt, 65
prompt
 PROM, 65

R

rebooting a system, 4

S

SBus, 59
SCSI target IDs, 7, 10
shutting down a system, 4
 with halt, 4
 with init, 4
 with shutdown, 4
 withreboot, 4
shutting down the system, 3
single-user
 booting to
 SPARC, 67
 continue to multiuser state
 SPARC, 67
Stop-A command, 68
system file
 x86, 68

T

tape drive
 displaying status, 30
tar command, 37
tensioning a tape cartridge, 31
touch /reconfigure command, 8

U

ufsdump command, 34
ufsrestore command, 35
uname -rs command, 2

V

verifying the operating environment, 2
VME bus, 59
vol
 file system, 49
volcheck

- using to check for floppy in drive, 49
- Volume Management
 - adding new devices, 46
 - configuring, 46
- configuring, 46

