Solaris 8 System Administration Supplement

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

Part Number 806-6611-10
January 2001

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.

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 REPONDRE 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

Contents

Preface

The *Solaris 8 System Administration Supplement* describes new functionality in Solaris™ Update releases. The information here supplements or supersedes information in the previous releases of Solaris 8 documentation sets. Solaris documentation is available on the Solaris 8 Documentation CD included in this release.

Note - The Solaris operating environment runs on two types of hardware, or platforms: SPARC™ and IA (Intel Architecture). The Solaris operating environment also runs on both 64–bit and 32–bit address spaces. The information in this document pertains to both platforms and address spaces unless called out in a special chapter, section, note, bullet, figure, table, example, or code example.

Ordering Sun Documents

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at `http://www1.fatbrain.com/documentation/sun`.

Accessing Sun Documentation Online

The docs.sun.com^{SM} Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Typographic Conventions

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

TABLE P–1 Typographic Conventions

Typeface or Symbol	**Meaning**	**Example**
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words, or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

Shell Prompts in Command Examples

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

TABLE P–2 Shell Prompts

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

CHAPTER **1**

What's New at a Glance

This chapter highlights new features added to the Solaris 8 Update releases.

Note - For the most current man pages, use the `man` command. The Solaris 8 Update release man pages include new feature information not found in the *Solaris 8 Reference Manual Collection*.

TABLE 1–1 Solaris 8 Features

Description	First Released
Managing Information	
Solaris Management Console™ (SMC) 2.0 is a GUI-based "umbrella application" that serves as the launching point for a variety of management tools. The SMC comes complete with a default toolbox containing the following tools: ■ Processes – Suspend, resume, monitor, and control processes. ■ Users – Set up and maintain user accounts, user templates, groups, mailing lists, Administrative Roles, and Rights. Grant or deny rights to users and to administrative roles – to control the specific applications each can work with and which tasks each user can perform. ■ Scheduled Jobs – Schedule, start, and manage jobs. ■ Mounts and Shares — view and manage mounts, shares, and usage information. ■ Disks – Create and view disk partitions. ■ Serial Ports – Configure and manage existing serial ports. ■ Log Viewer – View application and command line messages and manage log files. You can also manage diskless clients, but with commands only, not through the GUI. You can add or delete tools from the default toolbox or create a new toolbox to manage a different set of tools by using the SMC Toolbox Editor. For more information about using the command line interface, see “Solaris Management Console Overview” on page 17. For information on how to start SMC, see “Starting Solaris Management Console” on page 19. Also see the help associated with each tool.	1/01
Web-based enterprise management (WBEM) includes standards for web-based management of systems, networks, and devices on multiple platforms. This standardization enables system administrators to manage desktops, devices, and networks. ■ In the 10/00 Update release, additions include a description of the system properties that the CIM Object Manager uses and descriptions of the new `Solaris_Printer` and other printing definition classes. ■ In the 1/01 Update release additions include: ▪ Updated description of `init.wbem` command, which now starts the Solaris Management Console (SMC) server as well as the CIM Object Manager. ▪ Added section that describes how to upgrade the CIM Object Manager repository. ▪ Updated Security chapter refers the user to Solaris Management Console (SMC) for implementing role-based access control (RBAC). ▪ `wbemlogviewer` application replaced with Solaris Management Console (SMC) Log Viewer for viewing log file information. ▪ Added description of new `Solaris_Network1.0.mof` file and `Solaris_Users1.0.mof` file. To see a list of changes to the WBEM book, see “Changes to *Solaris WBEM Services Administrator’s Guide*” on page 20 or to view the book, see *Solaris WBEM Services Administrator’s Guide.*	10/00 and updated in 1/01
Managing Servers and Clients	

TABLE 1–1 Solaris 8 Features *(continued)*

Description	First Released
Diskless client is available through the Solaris Management Console command line. You can manage diskless clients, list OS services for diskless client, and manage patches on all existing diskless clients. For information on diskless clients, see Chapter 3.	1/01
Managing System Security	
The role-based access control (RBAC) databases can now be managed through the Solaris Management Console (SMC) graphical interface. Rights can now contain other rights. Rights can now be assigned by default in the `policy.conf` file. For more information, see "Role-Based Access Control" on page 23.	1/01
The Generic Security Services Application Programming Interface (GSS-API) is a security framework that enables applications to protect the data they transmit. The GSS-API provides authentication, integrity, and confidentiality services to applications. The interface permits those applications to be entirely generic with respect to security. That is, they do not have to know the underlying platform (such as the Solaris platform) or security mechanism (such as Kerberos) being used. This means that applications that use the GSS-API can be highly portable. For more information, see *GSS-API Programming Guide*.	6/00
SPARC: *Solaris Smart Cards Administration Guide* has been updated. Information on setting up internal card readers has been added. Step-by-step instructions for setting up smart card support have been streamlined to make setting up smart cards easier. To see more changes in this book, see "SPARC: Changes to *Solaris Smart Cards Administration Guide*" on page 26. To view this book, see *Solaris Smart Cards Administration Guide*.	1/01
Managing Networks	
IP network multipathing provides your system with recovery from single-point failures with network adapters and increased traffic throughput. If a failure occurs in the network adapter, and if you have an alternate adapter connected to the same IP link, the system switches all the network accesses automatically from the failed adapter to the alternate adapter. This process ensures uninterrupted access to the network. Also, when you have multiple network adapters connected to the same IP link, you achieve increased traffic throughput by spreading the traffic across multiple network adapters. For more information, see *IP Network Multipathing Administration Guide*.	10/00

TABLE 1–1 Solaris 8 Features *(continued)*

Description	First Released
Mobile IP (Internet Protocol) enables the transfer of information to and from mobile computers, such as laptops and wireless communications. The mobile computer can change its location to a foreign network and still access and communicate with and through the mobile computer's home network. The Solaris implementation of Mobile IP supports only IPv4. For more information, see *Mobile IP Administration Guide*.	6/00
SPARC: Lightweight Directory Access Protocol (LDAP) is now supported in the iPlanet™ Web Server directory server. To set up the iPlanet directory server to support Solaris clients, see *LDAP Setup and Configuration Guide*.	1/01
Managing File Systems	
The performance of direct I/O, which is used by database applications to access unbuffered file-system data, has been improved by allowing concurrent read access and write access to regular UFS files. For more information on direct I/O concurrency, see "Improved UFS Direct I/O Concurrency" on page 39.	1/01
UFS Snapshots provides the new `fssnap` command for backing up a file system while the file system is mounted. A snapshot is a temporary image of a file system intended for backup operations. Previously, we recommended when using the `ufsdump` command, you bring the system to single-user mode to keep the file system inactive during a backup. For more information on UFS Snapshots, see "Creating UFS Snapshots" on page 41.	1/01
The `mkfs` command has been updated to improve performance when you create file systems. Improved `mkfs` performance is often 10 times faster than in previous Solaris releases. Performance improvements are seen on systems when creating both large and small file systems. However, the biggest `mkfs` performance improvements are seen on systems with high-capacity or high-speed disks.	1/01
Managing Removable Media	
Removable media management now fully supports removable media such as DVD-ROMs, Zip drives, Jaz drives, CD-ROMs, and diskettes. For information on how to use this feature, see Chapter 6.	6/00 Updated in 10/00
Managing Devices	
Improved dynamic reconfiguration error messages are intended to help system administrators troubleshoot problems when they remove a system resource such as a configured swap area or a dedicated dump device. For more information on dynamic reconfiguration, see "New Dynamic Reconfiguration Error Messages" on page 59.	1/01

TABLE 1–1 Solaris 8 Features *(continued)*

Description	First Released
You can use Solaris Print Manager to set up a Universal Serial Bus (USB) printer that is attached to a SPARC system with USB ports. For more information, see "SPARC: USB Printer Support" on page 73.. For an overview of USB, see "USB Overview" on page 61.	10/00 Updated in 1/01
Managing System Resources	
Extended accounting introduces a new variable-length, general-purpose accounting file format that represents general groups of accounting data. Also included is the ability to configure resource utilization that was recorded by the kernel in the various accounting files. For information on how to use this feature, see "Extended Accounting Features" on page 75.	6/00
Managing System Performance	
The enhanced directory name look-up cache (DNLC) improves performance when you access files in large directories. For information on how to use this feature, see "DNLC Improvements" on page 77.	6/00
Solaris Tunable Parameters Reference Manual has been updated. Information on the *semsys:seminfo_semmnu* parameter has been added to this book. To view the book, see *Solaris Tunable Parameters Reference Manual*.	1/01
Early Access	
This release includes an Early Access (EA) directory with EA software. For more information, see the Readme on the Solaris Software CD 2 of 2.	

CHAPTER **2**

Managing Information

The Solaris 8 software release includes new management tools. See the following sections for more information.

- “Solaris Management Console Overview” on page 17
- “Starting Solaris Management Console” on page 19
- “Changes to *Solaris WBEM Services Administrator’s Guide*” on page 20

Note - For the most current man pages, use the `man` command. The Solaris 8 Update release man pages include new feature information not found in the *Solaris 8 Reference Manual Collection*.

Solaris Management Console Overview

Solaris Management Console is new in the 1/01 release.

Solaris Management Console (SMC) 2.0 is a GUI-based "umbrella application" that serves as the launching point for a variety of management tools. The SMC comes complete with a default toolbox that contains the following tools:

- Processes – Suspend, resume, monitor, and control processes.
- Users — Set up and maintain user accounts, user templates, groups, mailing lists, Administrative Roles, and Rights. Grant or deny rights to users and to administrative roles–to control the specific applications each can work with and which tasks each user can perform.
- Scheduled Jobs – Schedule, start, and manage jobs.
- Mounts and Shares– View and manage mounts, shares, and usage information.

- Disks – Create and view disk partitions.
- Serial Ports – Configure and manage existing serial ports.
- Log Viewer – View application and command-line messages and manage log files.

Note - You can also manage diskless clients, but with commands only, not through the GUI. See "Using the Command Line Interface."

You can add or delete tools from the default toolbox, or create a new toolbox to manage a different set of tools by using the SMC Toolbox Editor.

For more information about starting SMC, see “Starting Solaris Management Console” on page 19. Also see the help associated with each tool.

Using the Command Line Interface

In addition to working with the GUI-based SMC, you can use the command line interface to:

- Populate security-attribute databases in a name service —`smattrpop.1m`
- Start the SMC – `smc.1m`
- Configure the SMC – `smcconf.1m`

You can also use commands to manage the following:

- Jobs in the crontab database – `smcron.1m`
- Diskless clients:
  - Available only through the command line, not the GUI – `smdiskless.1m`
  - Also list OS services for diskless clients and manage patches on all existing diskless clients – `smosservice.1m`

  For more information on diskless clients, see Chapter 4
- Entries in the `exec_attr` database – `smexec.1m`
- Group entries – `smgroup.1m`
- Email alias entries – `smmaillist.1m`
- Batch user operations – `smmultiuser.1m`
- OS services – `smosservice.1m`
- Profiles (rights) in the `prof_attr` and `exec_attr` databases – `smprofile.1m`
- Roles and users in role accounts – `smrole.1m`
- User entries – `smuser.1m`

For more information about each command, see the individual man page.

Starting Solaris Management Console

The Solaris Management Console (SMC) has three primary components:

- The Console
- The SMC Toolbox Editor
- The SMC server

The Console can be started from the command line (described in the following), from the Tools menu of the CDE front panel, or by double-clicking an SMC icon in Applications Manager or File Manager.

▼ To Start the Console From the Command Line

1. **From**`/usr/sadm/bin` **(by default), type:**

```
% smc
```

Note - You can start SMC as a normal user, but some tools or applications might not load unless you log in as root or you assume a role during SMC server login.

▼ To Start the SMC Toolbox Editor

1. **From** `/usr/sadm/bin` **(by default), type:**

```
% smc edit
```

Note - You can start the SMC Editor as a normal user, but you will not be able to save a server toolbox unless you log in as root.

▼ To Determine if the SMC Server Is Running

If you have trouble running SMC, it might be that the SMC server is not running or is somehow in a problem state. To determine if the SMC server is running, do the following:

1. **As root, type:**

```
# /etc/init.d/init.wbem status
```

If the SMC server is running, you should get a response like the following:

```
SMC server version 2.0.0 running on port 898
```

▼ To Start the SMC Server

1. **As root, type:**

```
# /etc/init.d/init.wbem start
```

After a short time a message should return: "SMC server started."

▼ To Stop the SMC Server

1. **As root, type:**

```
#/etc/init.d/init.wbem stop
```

A message should return: "SMC stopped."

Changes to *Solaris WBEM Services Administrator's Guide*

Web-Based Enterprise Management (WBEM) includes standards for web-based management of systems, networks, and devices on multiple platforms. This standardization enables system administrators to manage desktops, devices, and networks. The following provides an overview of additional information from the 10/00 and 1/01 release of the *Solaris WBEM Services Administrator's Guide.*

TABLE 2–1 Changes to WBEM

The following is an overview of new information in the 10/00 Update release.	Appendix B was updated to include the following: ■ Description of the `Solaris_CIMOM1.0.mof` file. This file contains a description of all the system properties that the CIM Object Manager uses. ■ The `Solaris_Device1.0.mof` file was expanded to include the description of the new `Solaris_Printer` and other printing definition classes, and the `Solaris_TimeZone` class. For details refer to the Solaris_Schema appendix.
The following is an overview of new information provided in the 1/01 Update release.	■ Chapter 2, CIM Object Manager, was updated as follows: ▪ Updated description of `init.wbem` command, which now starts the Solaris Management Console (SMC) server as well as the CIM Object Manager. ▪ Added section that describes how to upgrade the CIM Object Manager repository. ■ Chapter 3, Administering Security, is updated to refer the user to Solaris Management Console (SMC) for implementing role-based access control (RBAC). ■ Chapter 5, System Logging, was updated as follows: ▪ `wbemlogviewer` application replaced with Solaris Management Console (SMC) Log Viewer. ■ Appendix B, The Solaris Schema was updated as follows: ▪ The `Solaris_SerialPortSetting` class moved from the `Solaris_Core1.0.mof` file to the `Solaris_Device1.0.mof` file. ▪ Description of the `Solaris_Application1.0.mof` file was changed, which removed this package attribute: `Package Status`. One patch attribute was added: `Packages`. ▪ Description of the `Solaris_System1.0.mof` file updated, which added a complete list of classes that were defined in this file. ▪ Added description of new `Solaris_Network1.0.mof` file. ▪ Added description of new `Solaris_Users1.0.mof` file.

CHAPTER **3**

Managing Security

The Solaris 8 software release includes new security functionality. See the following sections for more information.

- "Role-Based Access Control" on page 23
- "*GSS-API Programming Guide*" on page 26
- "SPARC: Changes to *Solaris Smart Cards Administration Guide*" on page 26

Note - For the most current man pages, use the `man` command. The Solaris 8 Update release man pages include new feature information not found in the *Solaris 8 Reference Manual Collection*.

Role-Based Access Control

This functionality has been updated in the 1/01 software release.

The role-based access control (RBAC) databases can now be managed through the User tool in the Solaris Management Console (SMC) graphical interface. For more information on SMC, see "Solaris Management Console Overview" on page 17. The updated RBAC has the following changes.

- A terminology change has made obsolete: the term *execution profiles*. The term has been replaced with *rights profiles*, also referred to as *rights* (in the graphical interface) and *profiles* (on the command line and in files).
- In addition to authorizations and commands with security attributes, a rights profile can now include other rights profiles. If the same command appears in more than one subordinate rights profile, the first occurrence in the file takes precedence.

- The policy.conf(4) file now recognizes the keyword PROFS_GRANTED, which lets you assign rights profiles by default.

The following figure illustrates how the extended user attributes are supplied to the user.

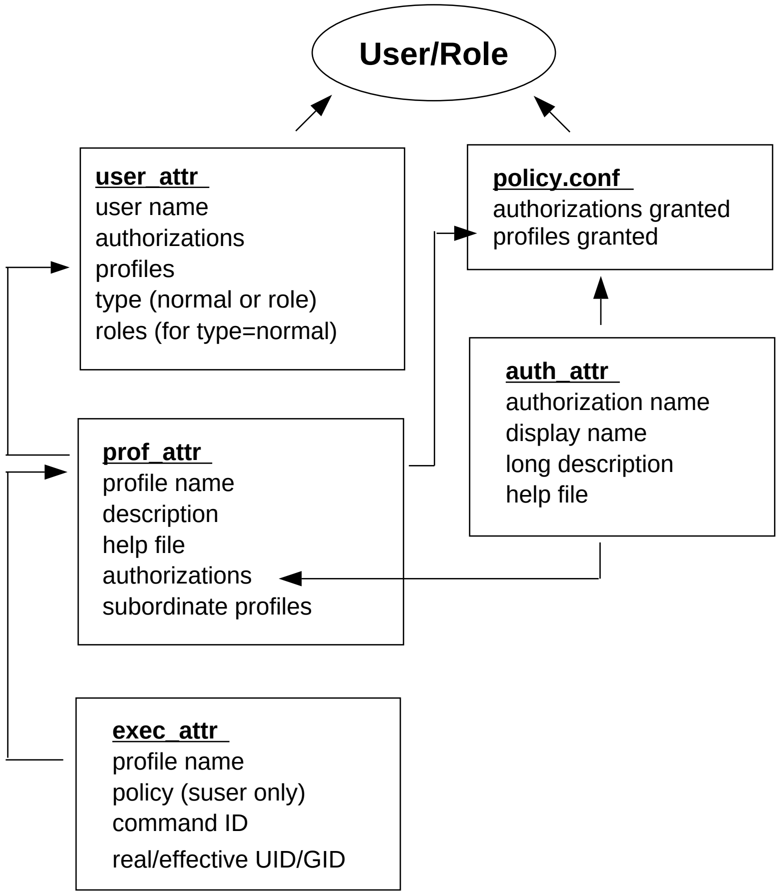

Figure 3–1 Extended Attribute Databases

The user_attr database contains the attributes that are shown and includes a comma-separated list of profile names. The contents of the profiles are split between the prof_attr file, which contains profile identification information, authorizations assigned to the profile, and subordinate profiles, and the exec_attr file, which identifies the policy and contains commands with their associated security attributes. The auth_attr file supplies authorization information to the SMC tools. Note that although you can assign authorizations directly to users through user_attr, this practice is discouraged. The policy.conf file supplies default attributes to be applied to all users.

For example, if the Printer Management rights profile is assigned to a user or role, the `user_attr` entry for that user or role contains the keyword/value pair: `profiles=Printer Management`. The `prof_attr` file defines this profile with the following line, which also specifies the help file and authorizations:

```
Printer Management:::Manage printers, daemons,
spooling:help=RtPrntAdmin.html;auths=solaris.admin.printer, /
solaris.admin.printer.modify,solaris.admin.printer.delete
```

In the `exec_attr` file, the following line assigns an effective user ID = `lp` to the command `/usr/sbin/accept` within the Printer Management profile:

```
Printer Management:suser:cmd:::/usr/sbin/accept:euid=lp
```

The following table lists commands that use authorizations.

TABLE 3–1 Commands and Their Authorizations

Command	Associated Authorizations
`at(1)`	solaris.jobs.user
`atq(1)`	solaris.jobs.admin
`crontab(1)`	solaris.jobs.user, solaris.jobs.admin
`allocate(1M)`	solaris.device.allocate, solaris.device.revoke
`deallocate(1M)`	solaris.device.allocate, solaris.device.revoke
`list_devices(1M)`	solaris.device.revoke
`rdate(1M)`	solaris.system.date
`smcron(1M)`	solaris.jobs.admin, solaris.jobs.user
`smdiskless(91M)`	solaris.admin.dcmgr.clients, solaris.admin.dcmgr.read
`smexec(1M)`	solaris.profmgr.read, solaris.profmgr.write
`smgroup(1M)`	solaris.admin.usermgr.read, solaris.admin.usermgr.write
`smmultiuser(1M)`, `smuser(1M)`	solaris.admin.usermgr.pswd, solaris.admin.usermgr.read, solaris.admin.usermgr.write, solaris.profmgr.assign, solaris.profmgr.delegate, solaris.role.assign, solaris.role.delegate

TABLE 3–1 Commands and Their Authorizations *(continued)*

Command	Associated Authorizations
`smmaillist(1M)`	solaris.admin.usermgr.read, solaris.admin.usermgr.write
`smosservice(91M)`	solaris.admin.dcmgr.admin, solaris.admin.dcmgr.read
`smprofile(1M)`	solaris.profmgr.read, solaris.profmgr.write
`smrole(1M)`	solaris.admin.usermgr.pswd, solaris.admin.usermgr.read, solaris.admin.usermgr.write, solaris.profmgr.assign, solaris.profmgr.delegate, solaris.role.assign, solaris.role.delegate

GSS-API Programming Guide

This book has been updated for the 6/00 release.

The Generic Security Services Application Programming Interface (GSS-API) is a security framework that enables applications to protect the data they transmit. The GSS-API provides authentication, integrity, and confidentiality services to applications. The interface permits those applications to be entirely generic with respect to security. That is, they do not have to know the underlying platform (such as the Solaris platform) or security mechanism (such as Kerberos) being used. This means that applications that use the GSS-API can be highly portable.

For more information, see *GSS-API Programming Guide*.

SPARC: Changes to *Solaris Smart Cards Administration Guide*

Solaris Smart Cards Administration Guide has been updated with the following additional information for the 1/01 release.

- The chapter on setting up card readers now has information on setting up internal card readers.

- New chapters describe the tasks that are needed for smart-card setup and additional configuration tasks that might be needed if the default smart-card properties are not sufficient for your security environment.
- Previous technical inaccuracies have been corrected.

To view this book, see *Solaris Smart Cards Administration Guide*.

CHAPTER **4**

Managing Diskless Clients

The Solaris 8 software release includes new diskless client functionality.

Note - For the most current man pages, use the `man` command. The Solaris 8 Update release man pages include new feature information that is not in the *Solaris 8 Reference Manual Collection*.

Managing Diskless Clients Overview

For the 1/01 release, Diskless Client Management updates the Solstice AdminSuite™ 2.3 Diskless Client tool. The AdminSuite 2.3 Diskless Client tool is GUI based, whereas Diskless Client Management consists solely of a command-line interface.

The following are supported:

- SPARC architecture or IA OS servers that run the Solaris 8 1/01 operating environment.
- SPARC architecture diskless clients that run either the Solaris 8 1/01, Solaris 2.7, or Solaris 2.6 operating environments from their OS server.

A *diskless client* is a workstation that depends on an *OS server*, or *host*, for its operating system, software, and storage. A diskless client mounts its root (`/`), `/usr`, and other file systems from its OS server. A diskless client has its own CPU and physical memory and can process data locally. However, a diskless client cannot operate if it is detached from its network or if its OS server malfunctions. A diskless client generates significant network traffic because of its continual need to function across the network.

Working With Diskless Client Management

You use the command-line interface to work with the Diskless Client Management tool. By writing your own shell scripts and using the commands shown in Table 4–1, you can easily set up and manage your diskless client environment.

TABLE 4–1 Diskless Client Management Commands

Command	Subcommand	Task
`/usr/sadm/bin/smosservice`	`add`	Add OS services
	`delete`	Delete OS services
	`list`	List OS services
	`patch`	Manage OS service patches
`/usr/sadm/bin/smdiskless`	`add`	Add a diskless client to an OS server
	`delete`	Delete a diskless client from an OS server
	`list`	List the diskless clients on an OS server
	`modify`	Modify the attributes of a diskless client

You can obtain help on these commands in two ways:

- *Usage statements* – To display a usage statement, use the `–h` option after you type the command and subcommand. For example, to display the usage statement for `smdiskless add`, type `/usr/sadm/bin/smdiskless add –h`.
- *Man pages* – To view a man page, type `man` and the command name. For example, to display the man page for `smdiskless`, type `man smdiskless`.

User Rights

Users can make use of either a subset or all of the Diskless Client Management commands, according to the rights to which they are assigned. Table 4–2 lists the rights that are required to use the Diskless Client Management commands.

TABLE 4–2 Required Rights

Right	Command	Task

Basic Solaris User, Network Management	`smosservice list`
	`smosservice patch`
	`smdiskless list`
		List OS services
		List OS patches
		List diskless clients

Network Management	`smdiskless add`	Add diskless clients
System Administrator	All commands	

Disk Space Requirements

Before you begin to set up your diskless client environment, make sure you have the required space available for each partition that is created. Table 4–3 lists the disk space requirements for each Diskless Client Management partition.

TABLE 4–3 Disk Space Requirements

Partition	Required Space (MB)
`/export/Solaris_`*version*	10
`/export/exec`	800
`/export/share`	5

TABLE 4–3 Disk Space Requirements *(continued)*

Partition	Required Space (MB)
`/export/dump/`*diskless_client_name*	32
`/export/root/templates/Solaris_`*version*	30
`/export/root/clone/Solaris_`*version*`/`*machine_class*	30–60, depending on machine class
`/export/root/`*diskless_client_name* (clone of above)	30–60, depending on machine class
`/export/swap/`*diskless_client_name*	32
`/tftpboot/inetboot.`*machine_class*`.Solaris_`*version*	200 KB per *machine_class*`.Solaris_`*version*

How to Set Up Your Diskless Client Environment

1. **Choose where to start.**
   - If your system currently supports diskless clients that were created with the AdminSuite 2.3 Diskless Client tool, proceed to step 2.
   - If your system does not currently support diskless clients that were created with the AdminSuite 2.3 Diskless Client tool, proceed to step 4.

2. **Remove the existing AdminSuite 2.3 diskless clients by using the** `/usr/sadm/bin/admhostdel` **command.**

3. **Remove the existing AdminSuite 2.3 OS services by using the** `/usr/sadm/bin/admhostmod` **command.**

4. **Upgrade the machine(s) designated as the OS server(s) to the Solaris 8 1/01 operating environment.**

5. **To view Diskless Client error messages in the Solaris Management Console (SMC) Log Viewer, start the SMC by typing the following at the command line:**

```
% /usr/sadm/bin/smc &
```

Then choose Log Viewer from the SMC main screen.

6. **Add the required OS services. See "Preparing to Add OS Services" on page 33 and "How to Add an OS Service" on page 35.**

7. **Add the diskless client(s). See "How to Add a Diskless Client" on page 35.**

8. **Boot each diskless client from the PROM level by using the** `boot net` **command. For more information on the** `boot net` **command, refer to *System Administration Guide, Volume 1*.**

Preparing to Add OS Services

When you use the `smosservice add` command to add OS services, you must type the *platform*, *mediapath*, and *cluster* of each diskless client platform that you want to support. Therefore, you must first do some high-level work to determine the following for each diskless client:

- *Platform* – You designate the diskless client platform in the format of *instruction_set*.*machine_class*.*Solaris_os_version*. For example, **`sparc.sun4u.Solaris_8`**. The following are the possible platform options:

instruction_set	***machine_class***	***Solaris_os_version***
sparc	sun4u sun4m sun4c sun4d	Solaris_8 Solaris_2.7 Solaris_2.6
i386	i86pc	Solaris_8 Solaris_2.7 Solaris_2.6

- *Media path* – The full path to the CD–ROM or network image that contains the operating system that you want to install for the diskless client. For example, `/net/install_files`.

Note - *If you are loading OS services from the Solaris 8 software CDs* – The Solaris 8 operating environment is delivered on multiple CDs. However, the Diskless Client Management software does not support this multiple CD distribution. You must run the scripts that are found on the Solaris 8 software CDs (and optional Language CD) to:

1. Create an install image on a server. For information on setting up an install server, refer to *Solaris 8 Advanced Installation Guide*.
2. Load the required OS services from the image.

The scripts are as follows:

- CD 1 of 2 – `/cdrom/cdrom0/s0/Solaris_8/Tools/setup_install_server`
- CD 2 of 2 – `/cdrom/cdrom0/s0/Solaris_8/Tools/add_to_install_server`
- Language CD – `/cdrom/cdrom0/s0/Solaris_8/Tools/add_to_install_server`

- *Cluster* – Depending on the configuration of the diskless client, you can specify one of four clusters that contain the Diskless Client functionality: `SUNWCXall`, `SUNWCall`, `SUNWCprog`, or `SUNWCuser`. You must use *the same cluster* for diskless clients that run the same operating environment on the same machine (SPARC or IA).

  For example, to set up the following diskless clients:

  - sparc.sun4m.Solaris_8
  - sparc.sun4u.Solaris_8
  - sparc.sun4d.Solaris_8

  You would specify the `SUNWCXall` cluster for each diskless client because the machine that runs sun4u requires `SUNWCXall`, and all diskless clients that run the same operating environment (in this situation, Solaris_8) on the same machine must use the same cluster.

Note - If you are using a sun4u machine, or if you are using a machine with an accelerated 8-bit color memory frame buffer (cgsix), you *must* specify `SUNWCXall` as the cluster.

How to Add an OS Service

After you determine the platform, media path, and cluster for each diskless client, you are ready to add OS services. The following directories are created and populated for each OS service that you add:

/export/Solaris_*version*/Solaris_*version_instruction_set*.all (symbolic link to
/export/exec/Solaris_*version*/Solaris_*version_instruction_set*.all)
/export/Solaris_*version*
/export/Solaris_*version*/var
/export/Solaris_*version*/opt
/export/share
/export/dump
/export/root/templates/Solaris_*version*
/export/root/clone
/export/root/clone/Solaris_*version*
/export/root/clone/Solaris_*version*/*machine_class*

1. **Use the** `/usr/sadm/bin/smosservice add` **command, including the required mediapath, platform, and cluster options, to add the first OS service.**

   For example, the following command adds an OS service for the Solaris 8 software for the sun4u machine class, where the OS server is not using a name service:

   ```
   % /usr/sadm/bin/smosservice add -- -x mediapath=/net/image/5.8/sparc \
   -x platform=sparc.sun4u.Solaris_8 -x cluster=SUNWCXall
   ```

   The installation process can require 45 minutes, depending on the server speed and the OS service configuration you choose.

2. **Continue to use the** `/usr/sadm/bin/smosservice add` **command to add any other OS services.**

3. **When you are finished adding OS services, use the** `/usr/sadm/bin/smosservice list` **command to verify that the OS services were installed.**

How to Add a Diskless Client

The following default directories are created and populated on the OS server for each diskless client that you add:

`/export/root/`*diskless_client_name*
`/export/swap/`*diskless_client_name*
`/export/dump/`*diskless_client_name*
`/tftpboot/`*diskless_client_ipaddress_in_hex*

Note - You can modify the default locations of the `/root`, `/swap`, and `/dump` directories by using the `–x` option. However, do not create these directories under the `/export` branch.

To add a diskless client:

1. **Use the** `smdiskless add` **command, including the required IP address, Ethernet address (MAC address), name, and operating system options, for the first diskless client that you want to add.**

   For example, the following adds the “client1” diskless client, which runs Solaris 8 on a sun4u machine:

```
% /usr/sadm/bin/smdiskless add -- -i 129.9.200.1 \
-e 8:0:11:12:13:14 -n client1 -x os=sparc.sun4u.Solaris_8 \
```

Note - The operating system is in the format of *instruction_set*.*machine_class*.*Solaris_os_version* and is equivalent to the *platform* you specified when you used the `smosservice` command to set up OS services.

---

2. **Continue to use the** `/usr/sadm/bin/smdiskless add` **command to add each remaining diskless client.**

3. **When you are finished adding diskless clients, use the** `/usr/sadm/bin/smdiskless list` **command to verify that the diskless clients were installed.**

Patching OS Services Overview

You use the `/usr/sadm/bin/smosservice patch` command to manage the OS service patches. You can:

- Establish an `/export/diskless/Patches` patch spool directory on an OS server.
- Add patches to the patch spool directory.
- Delete patches from the patch spool directory.

- List the patches in the patch spool directory.
- Synchronize spooled patches out to clients. (You must reboot each synchronized client for the client to recognize the patch update.)

Note - Diskless clients share the `/export/exec/Solaris_`*version*`/Solaris_`*version_instruction_set*`.all` directory on the OS server. Because this directory is shared, patches that you apply to this directory are not printed when you type the `showrev –p` command from the diskless client. However, kernel patches are applied to the diskless client's root directory, and therefore print when you type the `showrev –p` command from the diskless client.

How to Add Non–Solaris Patches

If you want to add non-Solaris patches to an OS server, do the following:

1. **Create a** `.copyof*` **"staging area" for the patches, using the following command:**

```
% /usr/sadm/bin/smosservice patch -- -m -U
```

2. **Add the patch(es) to the staging area using the following command:**

```
% /usr/sadm/bin/smosservice patch -- -a staging area
```

3. **Push the patch(es) out to the clients using the following command:**

```
% /usr/sadm/bin/smosservice patch -- -a -m -U
```

CHAPTER 5

Managing File-Systems

The Solaris 8 software release includes new UFS functionality. See the following sections for more information.

- "Improved UFS Direct I/O Concurrency" on page 39
- "UFS Snapshots Overview" on page 40

Note - For the most current man pages, use the man command. The Solaris 8 Update release man pages include new feature information that is not in the *Solaris 8 Reference Manual Collection*.

Improved UFS Direct I/O Concurrency

The UFS software has been enhanced in the Solaris 8 1/01 release. The information here adds to information on direct I/O that is in the "Managing File Systems" section of the "Managing File Systems (Overview)" in *System Administration Guide, Volume 1*.

The performance of direct I/O, which is used by database applications to access unbuffered file-system data, has been improved by allowing concurrent read and write access to regular UFS files. Previously, an operation that updates file data would lock out all other read or write accesses until the update operation was completed.

Concurrent writes are restricted to the special case of file rewrites. If the file is being extended, writing is single threaded as before. Generally, databases pre-allocate files and seldom extend them thereafter. Therefore, the effects of this enhancement are seen during normal database operations.

The direct I/O improvements bring I/O-bound database performance on a UFS file system to about 90% of raw partition access speeds. If the database is CPU bound or bus bandwidth bound, you might not see any improvement.

Consider running your I/O database applications with direct I/O enabled if you are already using UFS to store database tables. Use your database administrative procedures to enable direct I/O, if possible. If you cannot enable direct I/O through your database product, use the `mount –o forcedirectio` option to enable direct I/O for each file system or use the `directio(3C)` library call to enable direct I/O.

See `mount_ufs(1M)` or `directio(3C)` for more information.

UFS Snapshots Overview

The Solaris 8 1/01 release includes the new `fssnap` command for backing up file systems while the file system is mounted. The information here adds to information on backing up file systems that is in "Backing Up and Restoring File Systems (Overview)" in *System Administration Guide, Volume 1*.

You can use the `fssnap` command to create a read-only snapshot of a file system. A *snapshot* is a file system's temporary image that is intended for backup operations.

When the `fssnap` command is run, it creates a virtual device and a backing-store file. You can back up the *virtual device*, which looks and acts like a real device, with any of the existing Solaris backup commands. The *backing-store* file is a bit-mapped file that contains copies of pre-snapshot data that has been modified since the snapshot was taken.

Why Use UFS Snapshots?

UFS snapshots enables you to keep the file system mounted and the system in multiuser mode during backups. Previously, you were advised to bring the system to single-user mode to keep the file system inactive when using the `ufsdump` command to perform backups. You can also use additional Solaris backup commands like `tar` and `cpio` to back up a UFS snapshot for more reliable backups.

The `fssnap` command gives administrators of non-enterprise-level systems the power of enterprise-level tools like Sun StorEdge™ Instant image without the large storage demands.

UFS snapshots is similar to the Instant Image product. Instant Image allocates space equal to the size of the entire file system that is being captured. However, the backing-store file that was created by UFS snapshots occupies only as much disk space as needed, and you can place a maximum size on the backing-store file, if desired.

This table describes specific differences between UFS snapshots and Instant Image.

UFS Snapshots	Instant Image
Size of the backing-store file depends on how much data has changed since the snapshot was taken	Size of the backing-store file equivalent equals the size of the entire file system being copied
Does not persist across system reboots	Persists across system reboots
Works on UFS file systems	Cannot be used with root (`/`) or `/usr` file systems
Part of the Solaris 1/01 release	Part of the Enterprise Services Package

Although UFS snapshots can make copies of large file systems, Instant Image is better suited for enterprise-level systems. UFS snapshots is better suited for smaller systems.

UFS Snapshots Performance Issues

When the file-system snapshot is first created, users of the file system might notice a slight pause. The length of the pause increases with the size of the file system to be captured. While the file-system snapshot is active, users of the file system might notice a slight performance impact the file system is written to, but they will see no impact when reading the file system is read.

Creating UFS Snapshots

When you use the `fssnap` command to create a file-system snapshot, observe how much disk space the backing-store file consumes. The backing-store file uses no space, and then it grows quickly, especially on heavily used systems. Make sure the backing-store file has enough space to grow, or limit its size with the `–o maxsize=`*n* `[k,m,g]` option, where *n* `[k,m,g]` is the maximum size of the backing-store file.

Caution - If the backing-store file runs out of space, the snapshot might delete itself, which causes the backup to fail. Check the `/var/adm/messages` file for possible snapshot errors.

▼ How to Create a UFS Snapshot

1. **Become superuser.**

2. **Make sure there is enough disk space for the backing-store file.**

```
# df -k
```

3. **Make sure that a backing-store file of the same name and location does not already exist.**

```
# ls /file-system/backing-store-file
```

4. **Create the file-system snapshot.**

```
# fssnap -F ufs -o bs=/file-system/backing-store-file /file-system
```

Examples—Creating a UFS Snapshot

The following example creates a snapshot of the `/usr` file system. The backing-store file is `/scratch/usr.back.file`, and the virtual device is `/dev/fssnap/1`.

```
# fssnap -F ufs -o bs=/scratch/usr.back.file /usr
/dev/fssnap/1
```

The following example limits the backing-store file to 500 Mbytes.

```
# fssnap -F ufs -o maxsize=500m,bs=/scratch/usr.back.file /export/home
/dev/fssnap/1
```

▼ How to Display UFS Snapshot Information

You can display the current snapshots on the system by using the `fssnap –i` option. If you specify a file system, you see detailed information about that snapshot. If you don't specify a file system, you see information about all of the current file system snapshots and their corresponding virtual devices.

1. **Become superuser.**

2. **List current snapshots.**

```
# fssnap -i
   0    /
   1    /usr
```

To display detailed information about a specific snapshot, use the following:

```
# fssnap -i /usr
Snapshot number               : 1
Block Device                  : /dev/fssnap/1
Raw Device                    : /dev/rfssnap/1
Mount point                   : /usr
Device state                  : idle
Backing store path            : /scratch/usr.back.file
Backing store size            : 480 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Tue Aug 08 09:57:07 2000
Copy-on-write granularity     : 32 KB
```

Deleting a UFS Snapshot

When you create a UFS snapshot, you can specify that the backing-store file is unlinked, which means the backing-store file is removed after the snapshot is created. If you don't specify the `–o unlink` option when you create a UFS snapshot, you will have to delete it manually.

The backing-store file occupies disk space until the snapshot is deleted, whether you use the `–o unlink` option to remove the backing-store file or you remove it manually.

How to Delete a UFS Snapshot

You can delete a snapshot either by rebooting the system or by using the `fssnap -d` command and specifying the path of the file system that contains the file-system snapshot.

1. **Become superuser.**

2. **Identify the snapshot to be deleted.**

```
# fssnap -i
```

3. **Delete the snapshot.**

```
# fssnap -d  /file-system
Deleted snapshot 1.
```

4. **(Optional) If you did not use the** `–o unlink` **option when you created the snapshot, you need to delete the backing-store file manually.**

```
# rm /file-system/backing-store-file
```

Example—Deleting a UFS Snapshot

The following example deletes a snapshot and assumes that the `unlink` option was not used.

```
# fssnap -i
    0    /    1    /usr
# fssnap -d /usr
 Deleted snapshot 1.
# rm /scratch/usr.back.file
```

Backing Up a UFS Snapshot

The virtual device that contains the file-system snapshot acts as a standard read-only device. This means you can back up the virtual device as if you were backing up a file-system device.

If you are using the `ufsdump` command to back up a UFS snapshot, you can specify the snapshot name during the backup. See the following section for more information.

If you are using the `tar` command to back up the snapshot, mount the snapshot before backing it up, like this:

```
# mkdir /backups/home.bkup
# mount -F UFS -o ro /dev/fssnap/1 /backups/home.bkup
# cd /backups/home.bkup
# tar cvf /dev/rmt/0 .
```

For more information on how to back up a file system see “Backing Up Files and File Systems (Tasks)” in *System Administration Guide, Volume 1*.

How to Back Up a UFS Snapshot

1. **Become superuser.**

2. **Identify the file-system snapshot to be backed up.**

```
# fssnap -i /file-system
```

   For example:

```
# fssnap -i /usr
Snapshot number               : 1
Block Device                  : /dev/fssnap/1
Raw Device                    : /dev/rfssnap/1
Mount point                   : /usr
Device state                  : idle
Backing store path            : /scratch/usr.back.file
Backing store size            : 480 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Tue Aug 08 09:57:07 2000
Copy-on-write granularity     : 32 KB
```

3. **Back up the file system snapshot.**

```
# ufsdump 0ucf /dev/rmt/0 /snapshot-name
```

   For example:

```
# ufsdump 0ucf /dev/rmt/0 /dev/rfssnap/1
```

4. **Verify the snapshot is backed up.**

```
# ufsrestore ta /dev/rmt/0
```

▼ How to Create an Incremental Dump of a UFS Snapshot

If you want to create a file-system snapshot incrementally, which means only the files that have been modified since the last snapshot are backed up, use the `ufsdump` command with the new `N` option. This option specifies the file-system device name to be inserted into the `/etc/dumpdates` file for tracking incremental dumps.

The following `ufsdump` command specifies an embedded `fssnap` command to create an incremental dump of a file system.

1. **Become superuser.**

2. **Create an incremental dump of a file system snapshot.**

```
# ufsdump 1ufN /dev/rmt/0 /dev/rdsk/c0t1d0s0 'fssnap -F ufs -o raw,bs=
/export/scratch,unlink /dev/rdsk/c0t1d0s0'
```

   The –`o raw` option is used in the example to display the name of the raw device instead of the block device. By using this option, you make it easier to embed the `fssnap` command in commands that require the raw device instead, such as the `ufsdump` command.

3. **Verify the snapshot is backed up.**

```
# ufsrestore ta /dev/rmt/0
```

Restoring Data From a UFS Snapshot Backup

The backup created from the virtual device is essentially just a backup of what the original file system looked like when the snapshot was taken. When you restore from the backup, restore as if you had taken the backup directly from the original file system, such as one that used the `ufsrestore` command. For more information on restoring file systems see "Restoring Files and File Systems (Tasks)" in *System Administration Guide, Volume 1.*

CHAPTER **6**

Managing Removable Media

The Solaris 8 software release includes new removable media functionality.

Note - For the most current man pages, use the `man` command. The Solaris 8 Update release man pages include new feature information that is not in the *Solaris 8 Reference Manual Collection*.

Managing Removable Media Overview

This functionality is updated for the Solaris 8 6/00 and the documentation updated again in the 10/00 software release.

Volume management features have been improved in the Solaris 8 6/00 software release to fully support removable media. This improvement means that DVD-ROMs, Iomega and Universal Serial Bus (USB) Zip drives and Jaz drives, CD-ROMs, and diskettes are mounted and available for reading when they are inserted.

Both the Common Desktop Environment (CDE) volume management and Solaris command-line features have been updated in this release. The information here adds to information on managing removable media in “Guidelines for Using CDs and Diskettes (Overview)” in *System Administration Guide, Volume 1*. For information on using File Manager to administer this feature, see “Using Removable Media Manager” in *Solaris 8 User Supplement*.

With the volume management improvements, you can:

- Format, label, and set read or write software protection on removable media with the new `rmformat` command. This command replaces the `fdformat` command for formatting removable media.

- Create and verify a PCFS file system on removable media with the `mkfs_pcfs` and `fsck_pcfs` commands.
- Create an `fdisk` partition and a PCFS file system on removable media on a SPARC system to facilitate data transfers to IA systems.

Guidelines for using removable media are:

- Use UDFS and PCFS to transfer data between DVD media.
- Use the `tar` or `cpio` commands to transfer files between rewritable media such as a PCMCIA memory card or diskette with a UFS file system. A UFS file system that is created on a SPARC system is not identical to a UFS file system on PCMCIA or to a diskette that is created on an IA system.
- Set write protection to protect important files on Jaz or Zip drives or diskettes. Apply a password to Iomega media.

Accessing Information on Removable Media

You can access information on removable media with or without using volume manager. For information on accessing information on removable media with File Manager, see "Using Removable Media Manager" in *Solaris 8 User Supplement*.

Starting in the Solaris 8 6/00 release, volume manager (`vold`) actively manages all removable media devices. This means any attempt to access removable media with device names such as `/dev/rdsk/c`*n*`t`*n*`d`*n*`s`*n* or `/dev/dsk/c`*n*`t`*n*`d`*n*`s`*n* will be unsuccessful.

By using CDE's Removable Media Manager or the volume manager path names such as `/cdrom0`, `/floppy`, `/rmdisk`, `/jaz0`, or `/zip0`, you can access the devices when the volume manager, `vold`, is running.

You can also access removable media by their entries in the `/vol/dev` directory. For example:

`/vol/dev/rdiskette0/`*volume-name*

for a diskette, or

`/vol/dev/rdsk/c`*n*`t`*n*`d`*n*`/`*volume-name*

for a CD-ROM or removable hard disk.

If a removable media device contains a removable medium, its alias appears in the `/vol/dev/aliases` directory as a symbolic link to its path in the `/vol/dev` directory. For example, if a diskette that is labeled `test` is in diskette drive 0 and a CD that is labeled `test` is in the CD-ROM drive at `/dev/rdsk/c2t1d0`, you see the following output:

```
$ ls -l /vol/dev/aliases
lrwxrwxrwx 1 root root 30 May 11 12:58 cdrom0 -> /vol/dev/rdsk/c2t1d0/test
lrwxrwxrwx 1 root root 30 May 11 12:58 floppy0 -> /vol/dev/rdiskette0/test
```

If you are unsure which device name to choose, use the `eject -n` command to display device names for all removable media devices. For example, use the device name on the right side of `eject -n` output to determine which device name to use with the `fsck`, `mkfs`, or `newfs` commands.

How to Access Information on Removable Media

Use the appropriate device name to access information by using the command-line interface. You can use the volume manager's nickname from the command line by running the `volcheck` command before you access the removable media. See *rmformat(1)* for an explanation of device names.

Examples—Accessing Information on Removable Media

To access information on a diskette, use:

```
$ volcheck
$ ls /floppy
myfile
```

To access information on a Jaz drive, use:

```
$ volcheck
$ ls /rmdisk
jaz0/  jaz1/
```

To access information on a CD-ROM, use:

```
$ volcheck
$ ls /cdrom
solaris_8_sparc/
```

Accessing Jaz Drives or Zip Drives

You can determine whether accessing your Jaz or Zip drives changes from previous Solaris releases, depending on whether you upgrade or install the Solaris 8 6/00 release:

- If you are upgrading to the Solaris 8 6/00 release from a previous Solaris release, you can continue to access your Jaz drives and Zip drives in the same way as in previous releases.
- If you are freshly installing the Solaris 8 6/00 release, you cannot access your Jaz drives and Zip drives in the same way as in previous Solaris releases.

Follow the next procedure if you want to access your Jaz and Zip drives the same way as in previous Solaris releases.

1. **Become superuser.**
2. **Comment the following line in the `/etc/vold.conf` file by inserting a pound (#) sign at the beginning of the text, like this:**

```
# use rmdisk drive /dev/rdsk/c*s2 dev_rmdisk.so rmdisk%d
```

3. **Reboot the system.**

```
# init 6
```

Formatting Removable Media (`rmformat`)

You can use the `rmformat` command to format removable media, including the following types of diskettes:

- Double-density – 720 Kbytes (3.5 inch)
- High-density – 1.44 Mbytes (3.5 inch)

The `rmformat` command is a non-superuser utility that can format and protect rewritable removable media. The `rmformat` command has three formatting options:

- `quick` – This option formats removable media without certification or with limited certification of certain tracks on the media.
- `long` – This option formats removable media completely. For some devices using this option might include the certification of the whole media by the drive itself.
- `force` – This option formats completely without user confirmation. For media with a password-protection mechanism, this option clears the password before

formatting. This feature is useful when a password is forgotten. On media without password protection, this option forces a long format.

How to Format Removable Media (`rmformat`)

The `rmformat` command formats the media and by default creates two partitions on the media: partition 0 and partition 2 (the whole media).

1. **Verify that the volume manager is running, which means you can use the shorter nickname for the device name.**

```
$ ps -ef | grep vold
root   212     1  0   Nov 03 ?        0:01 /usr/sbin/vold
```

See *System Administration Guide, Volume 1* for information on determining removable media device names and starting volume manager if it is not running.

2. **Format the removable media.**

```
$ rmformat -F [ quick | long | force ] device-name
```

See the previous section for more information on `rmformat` formatting options.

If the `rmformat` output indicates bad blocks, see the following procedure for repairing bad blocks.

3. **(Optional) Label the removable media with an 8-character label to be used in the Solaris environment.**

```
$ rmformat -b label device-name
```

See `mkfs_pcfs(1M)` for information on creating a DOS label.

Examples—Formatting Removable Media

This example formats a diskette.

```
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
.........................................................................
```

This example formats a Zip drive.

```
$ rmformat -F quick zip0
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
.........................................................................
```

How to Format Removable Media for a UFS or UDFS File System

1. **Format the media.**

   ```
   $ rmformat -F quick device-name
   ```

2. **(Optional) Create an alternate Solaris partition table.**

   ```
   $ rmformat -s slice-file  device-name
   ```

   A sample slice file looks like the following:

```
slices: 0 = 0, 30MB, "wm", "home" :
              1 = 30MB, 51MB :
              2 = 0, 94MB, "wm", "backup" :
              6 = 81MB, 13MB
```

   See *System Administration Guide, Volume 1* for information on creating an alternate Solaris partition table.

3. **Become superuser.**

4. **Determine the appropriate file-system type and select one of the following:**

   a. **Create a UFS file system.**

      ```
      # newfs device-name
      ```

   b. **Create a UDFS file system.**

      ```
      # mkfs -F udfs device-name
      ```

Example—Formatting Removable Media for a UFS File System

The following example formats a diskette and creates a UFS file system.

```
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
$ su
# newfs /dev/rdiskette
newfs: construct a new file system /dev/rdiskette: (y/n)? y
/dev/rdiskette: 2880 sectors in 80 cylinders of 2 tracks, 18 sectors
        1.4MB in 5 cyl groups (16 c/g, 0.28MB/g, 128 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 640, 1184, 1792, 2336,
#
```

▼ How to Format Removable Media for a PCFS File System

1. **Format the removable media.**

   ```
   $ rmformat -F quick device-name
   ```

2. **Become superuser.**

3. **(Optional) Create an alternate Solaris `fdisk` partition table.**

   ```
   # fdisk device-name
   ```

   See *System Administration Guide, Volume 1* for information on creating an `fdisk` partition.

4. **Create a PCFS file system.**

   ```
   # mkfs -F pcfs device-name
   ```

Examples—Formatting Removable Media for a PCFS File System

This example includes how to create an alternate `fdisk` partition.

```
$ rmformat -F quick /dev/rdsk/c0t4d0s2:c
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
$ su
# fdisk /dev/rdsk/c0t4d0s2:c
# mkfs -F pcfs /dev/rdsk/c0t4d0s2:c
Construct a new FAT file system on /dev/rdsk/c0t4d0s2:c: (y/n)? y
#
```

This example describes how to create a PCFS file system without an `fdisk` partition.

```
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
$ su
# mkfs -F pcfs -o nofdisk,size=2 /dev/rdiskette
Construct a new FAT file system on /dev/rdiskette: (y/n)? y
#
```

▼ How to Check a PCFS File System on Removable Media

1. **Become superuser.**

2. **Check the PCFS file system.**

   ```
   # fsck -F pcfs device-name
   ```

Example—Checking a PCFS File System on Removable Media

```
# fsck -F pcfs /dev/rdsk/c0t4d0s2
** /dev/rdsk/c0t4d0s2
** Scanning file system meta-data
** Correcting any meta-data discrepancies
1457664 bytes.
0 bytes in bad sectors.
0 bytes in 0 directories.
0 bytes in 0 files.
1457664 bytes free.
512 bytes per allocation unit.
2847 total allocation units.
2847 available allocation units.
#
```

How to Repair Bad Blocks on Removable Media

You can only use the `rmformat` command to verify, analyze, and repair bad sectors that are found during verification if the drive supports bad block management. Most diskettes and PCMCIA memory cards do not support bad block management.

If the drive supports bad block management, a best effort is made to rectify the bad block. If the bad block cannot be rectified despite the best effort mechanism, a message indicates a failure to repair.

1. **Repair bad blocks on removable media.**

   ```
   $ rmformat -c block-numbers device-name
   ```

   Supply the block number in decimal, octal, or hexadecimal format from a previous `rmformat` session.

2. **Verify the media.**

   ```
   $ rmformat -V read device-name
   ```

Applying Read or Write and Password Protection to Removable Media

You can apply read protection or write protection and set a password on Iomega media such as Zip drives and Jaz drives. For other types of media, you can enable or disable write protection without a password.

How to Enable or Disable Write Protection on Removable Media

1. **Determine whether you want to enable or disable write protection and select one of the following:**

   a. **Enable write protection.**

      ```
      $ rmformat -w enable device-name
      ```

   b. **Disable write protection.**

```
$ rmformat -w disable device-name
```

2. **Verify whether the media's write protection is enabled or disabled.**

```
$ rmformat -p device-name
```

How to Enable or Disable Read or Write Protection and a Password on Iomega Media

You can apply a password with a maximum of 32 characters for Iomega media that support this feature. You cannot set read protection or write protection without a password on Iomega media. In this situation, you are prompted to provide a password.

You receive a warning message if you attempt to apply a password on media that does not support this feature.

1. **Determine whether you want to enable or disable read protection or write protection and a password.**

   a. **Enable read protection or write protection.**

```
$ rmformat -W enable device-name
Please enter password (32 chars maximum): xxx
Please reenter password:
```

```
$ rmformat -R enable device-name
Please enter password (32 chars maximum): xxx
Please reenter password:
```

   b. **Disable read protection or write protection and remove the password.**

```
$ rmformat -W disable device-name
Please enter password (32 chars maximum): xxx
```

```
$ rmformat -R disable device-name
Please enter password (32 chars maximum): xxx
```

2. **Verify whether the media's read or write protection is enabled or disabled.**

   ```
   $ rmformat -p device-name
   ```

Examples—Enabling or Disabling Read or Write Protection

This example enables write protection and sets a password on a Zip drive.

```
$ rmformat -W enable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
Please reenter password: xxx
```

This example disables write protection and removes the password on a Zip drive.

```
$ rmformat -W disable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
```

This example enables read protection and sets a password on a Zip drive.

```
$ rmformat -R enable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
Please reenter password: xxx
```

This example disables read protection and removes the password on a Zip drive.

```
$ rmformat -R disable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
```

CHAPTER **7**

Managing Devices

The Solaris 8 software release includes new device functionality. See the following sections for more information.

- "New Dynamic Reconfiguration Error Messages" on page 59
- "USB Overview" on page 61
- "SPARC: USB Printer Support" on page 73

Note - For the most current man pages, use the `man` command. The Solaris 8 Update release man pages include new feature information that is not in the *Solaris 8 Reference Manual Collection*.

New Dynamic Reconfiguration Error Messages

The dynamic reconfiguration software has been enhanced in the Solaris 8 1/01 release to improve troubleshooting dynamic reconfiguration problems. The information here adds to information on troubleshooting dynamic reconfiguration problems in "Configuring Devices" in *System Administration Guide, Volume 1*.

If you attempt to remove a system resource, such as a swap device or a dedicated dump device, with the `cfgadm` command, you will see error messages displayed if the system resource is still active.

The error messages are described in the following section.

Error Message

```
cfgadm: Component system is busy, try again: failed to
offline:
     device  path
          Resource              Information
     ------------------  --------------------------
     /dev/dsk/device-name   swap area
```

Cause

You attempted to remove or replace one or more configured swap areas.

Solution

Unconfigure the swap areas on the device that is specified and retry the `cfgadm` operation.

Error Message

```
cfgadm: Component
system is busy, try again: failed to offline:
     device  path
          Resource              Information
     ------------------  --------------------------
     /dev/dsk/device-name   dump device (swap)
```

Cause

You attempted to remove or replace a dump device that is configured on a swap area.

Solution

Unconfigure the dump device that is configured on the swap area and retry the `cfgadm` operation.

Error Message

```
cfgadm: Component
system is busy, try again: failed to offline:
     device  path
          Resource              Information
     ------------------  --------------------------
     /dev/dsk/device-name   dump device (dedicated)
```

Cause

You attempted to remove or replace a dedicated dump device.

Solution

Unconfigure the dump device that is dedicated and retry the `cfgadm` operation.

See `cfgadm(1M)` for more information.

USB Overview

This overview is revised in the Solaris 8 1/01 software release.

Universal Serial Bus (USB) was developed by the PC industry to provide a low-cost solution for attaching peripheral devices, such as keyboards, mouse devices, and printers, to a system.

USB connectors are designed to fit only one type of cable, one way. Devices can connect to hub devices, which connect several devices, including other hub devices. The primary design motivation for USB is to alleviate the need for multiple connector types for different devices, thereby reducing the clutter on the back panel of a system. Additional advantages of using USB devices are:

- USB devices are hot-pluggable. See "Hot-Plugging USB Devices" on page 72 for more information.
- Supports a maximum of 126 devices in the Solaris environment.
- Supports a maximum of 12 Mbit/sec data transfer.
- Supports low speed (1.5 Mbit/sec) and full speed (12 Mbit/sec) devices.
- The bus can be easily extended by adding low-cost external hubs. Hubs can be connected to hubs to form a tree topology.

Sun Microsystems support for USB devices includes the following:

- Sun Blade™ 100 and Sun Blade 1000 systems running the Solaris 8 10/00 release provide USB device support.
- Sun Ray™ systems also support USB devices.
- IA systems that run the Solaris 8 Intel Platform Edition provide USB support for keyboard and mouse devices and for certain mass-storage devices, such as Zip drives, is also available. See `scsa2usb(7D)` for more information.

This table provides a listing of specific USB devices that are supported in the Solaris environment.

These USB Devices ...	Are Supported on These Systems ...
Keyboards and mouse devices	SPARC systems with Sun USB support based on the ohci(7D) controller IA systems with a USB bus based on the uhci(7D) controller Only onboard USB controllers are supported. Plug-in host controller PCI cards are not supported.
Mass storage	SPARC and IA
Printers	SPARC
Hub	SPARC and IA

Commonly Used USB Acronyms

The following table describes the USB acronyms that are used in the Solaris environment. See `http://www.usb.org` for a complete description of USB components and acronyms.

Acronym	Definition
USB	Universal Serial Bus
USBA	Universal Serial Bus Architecture (Solaris)
USBAI	USBA Client Driver Interface (Solaris)
HCD	USB host controller driver

USB Bus Description

The USB specification, free of royalties, and openly available. The specification defines the electrical and mechanical interfaces of the bus and the connectors.

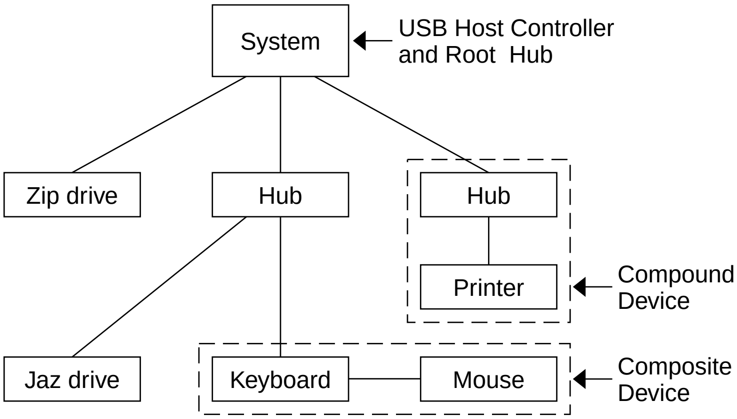

Figure 7–1 USB Physical Device Hierarchy

USB employs a topology in which hubs provide attachment points for USB devices. The host controller contains the root hub, which is the origin of all USB ports in the system. See "USB Host Controller and Root Hub" on page 66 for more information about hubs.

The previous example shows a system with three active USB ports. The first USB port has a Zip drive that does not have an embedded hub, so you cannot attach additional devices. The second USB port has a hub with a Jaz drive and a composite keyboard/mouse device connected. One of the ports from the secondary hub has a keyboard with an embedded hub where the mouse is attached.

The device tree path name for some of the devices displayed in the previous example are listed in this table.

Zip drive	`/pci@1f,4000/usb@5/storage@1`
Keyboard	`/pci@1f,4000/usb@5/hub@2/keyboard@1`
Mouse	`/pci@1f,4000/usb@5/hub@2/mouse@2`
Jaz Drive	`/pci@1f,4000/usb@5/hub@2/storage@3`
Printer	`/pci@1f,4000/usb@5/hub@3/printer@1`

USB Devices and Drivers

The USB devices are divided into device classes. Each device class has a corresponding driver. Devices within a class are managed by the same device driver. However, the USB specification also allows for vendor-specific devices that are not part of a specific class. Devices with similar attributes and services are grouped.

The Human Interface Device (HID) class contains devices that are user controlled such as keyboards, mouse devices, and joysticks. The Communication Device class contains devices that connect to a telephone, such as modems or an ISDN interface. Other device classes include the Audio, Monitor, Printer, and Storage Device classes. Each USB device contains descriptors that reflect the class of the device. A device class specifies how its members should behave with respect to configuration and data transfer. You can obtain additional class information from the `http://www.usb.org` site.

Solaris USB Architecture (USBA)

USB devices are represented as two levels of device tree nodes. A device node represents the entire USB *device* and one or more child *interface* nodes represent the individual USB interfaces on the device. For special cases, the device and interface nodes are *combined* into a single combined node.

Driver binding is achieved by using the compatible name properties. Refer to 3.2.2.1 of the IEEE 1275 USB binding and the *Writing Device Drivers* for more information. A driver can either bind to the entire device and control all the interfaces, or a driver can bind to just one interface, for example, a keyboard or mouse. If no vendor or class driver claims the entire device, a generic USB multi-interface driver is bound to the device level node. This driver attempts to bind drivers to each interface using compatible names properties defined in section 3.3.2.1 of the 1275 binding.

Figure 7–1 shows an example of a hub and printer as a *compound device*. Both the hub and the printer are enclosed in the same plastic case, but the hub and the printer have separate USB bus addresses. The same diagram shows an example of a *composite device*. The composite keyboard and controller are also enclosed in the same plastic case, but they have the same USB bus address. A cable connects the USB mouse to the composite keyboard/controller in this example.

The Solaris USB Architecture (USBA) adheres to the USB 1.0 and 1.1 specification plus Solaris driver requirements. The USBA model is similar to Sun Common SCSI Architecture (SCSA). The USBA is a thin layer that provides a generic USB transport layer abstraction to the client driver.

The differences between SCSA and USBA are that the SCSA relies on `.conf` files to probe the bus, while USB hub drivers are self-probing nexus drivers.

About USB in the Solaris Environment

The following section describes specific information you should know about USB in the Solaris environment.

USB Keyboards and Mouse Devices

Keep only one USB keyboard and mouse on the system at all times because multiple USB keyboards and mouse devices are not supported in the Solaris environment. See the following items for specific details.

- A keyboard and mouse connected anywhere on the bus are configured as console keyboard and mouse. It takes longer to boot the system if the keyboard and mouse are not on the root hub.
- You can move a console keyboard and mouse to another hub at any time *after* a system reboot. You cannot move the console keyboard and mouse *during* a reboot or at the `ok` prompt. After plugging in the keyboard and mouse, they are fully functional again.
- **SPARC only** – The power key on a USB keyboard behaves differently than the one on the Sun type 5 keyboard. On a USB keyboard, you can suspend or shut down the system by using the SUSPEND/SHUTDOWN key, but you cannot power-on the system.
- The left side of the keypad functionality is unavailable on non-Sun USB keyboards.
- Multiple keyboards are not supported:
  - The keyboards enumerate and are usable, but they are not plumbed as console keyboards.
  - The first keyboard that gets probed at boot time becomes the console keyboard. This might cause confusion if there are multiple keyboards plugged in at boot time.
  - If you unplug the console keyboard, the next available USB keyboard doesn't become the console keyboard. The next hot-plugged keyboard becomes the console keyboard.
- Multiple mouse devices are not supported:
  - The mouse devices enumerate and are usable, but they are not plumbed as console mouse devices.
  - The first mouse that is probed at boot time becomes the console mouse. This configuration might cause confusion if you have multiple mouse devices plugged in at boot time.
  - If you unplug the console mouse, the next available USB mouse doesn't become the console mouse. The next hot-plugged mouse becomes the console mouse.
- If you have a non–Sun (third-party) composite keyboard with a PS/2 mouse, and it is the first one to be probed, it becomes the console keyboard/mouse even if the PS/2 mouse is not plugged in. This means another USB mouse plugged into the system will not work because it is not configured as the console mouse.
- Only two- and three-button mouse devices are supported. A wheel-on-wheel mouse acts like a plain-button mouse. A mouse with more than three buttons functions like a three–button mouse.

USB Host Controller and Root Hub

A USB hub is responsible for:

- Monitoring the insertion or removal of a device on its ports
- Power-managing individual devices on its ports
- Controlling power to its ports

The USB host controller has an embedded hub called the *root hub*. The ports that are visible at the back panel are the ports of the root hub. The USB host controller is responsible for:

- Directing the USB bus. Individual devices cannot arbitrate for the bus.
- Polling the devices by using a polling interval determined by the device. The device is assumed to have sufficient buffering to account for the time between the polls.
- Sending data between the USB host controller and its attached devices. Peer-to-peer communication is not supported.

USB Hub Devices

- Do not cascade hubs beyond four levels on either SPARC or IA systems. On SPARC systems, the Open Boot PROM (OBP) cannot reliably probe beyond four levels of devices.
- Do not cascade bus-powered hubs. This means you cannot plug a bus-powered hub into another bus-powered hub. A bus-powered hub does not have its own power supply. A USB diskette device takes all its power from the bus and might not work on a bus-powered hub.

USB Storage Devices

Removable mass storage devices such as USB Zip, Jaz, Clik!, SmartMedia, CompactFlash, and ORB are supported, starting with the Solaris 8 10/00 release. See `scsa2usb(7D)` for a complete list of devices supported in the Solaris environment.

These devices can be managed with or without volume management. See `vold(1M)` for information on managing devices with volume management.

Managing USB Mass Storage Devices With `vold` Running

If you are running Solaris Common Desktop Environment (CDE), the USB removable mass storage devices are managed by the Removable Media Manager component of the CDE File Manager. See `dtfile(1)` for more information on the CDE File Manager.

Note - You must include the /usr/dt/man in your MANPATH variable to display the man pages listed in this section. You must also have /usr/dt/bin in your path and have CDE running to use these commands, or have a DISPLAY variable set to use these commands remotely.

The following table identifies the commands Removable Media Manager uses to manage storage devices from the CDE environment.

Command	Task
sdtmedia_format(1)	Format and label USB devices
sdtmedia_prop(1)	Display properties of the device
sdtmedia_prot(1)	Change device protection
sdtmedia_slice(1)	Create or modify slices on the device

After the USB device is formatted, it is usually mounted under the /rmdisk/*label* directory. See rmmount.conf(4) or vold.conf(4) for details on how to configure removable storage devices.

The following procedures describe how to manage USB mass storage devices with volume management. The device nodes are created under the /vol/dev directory. See scsa2usb(7D) for more information. The following procedures also describe how to add or remove hot-pluggable USB mass storage devices. Hot-plugging a device means the device is added or removed without shutting down the operating system or powering off the system.

How to Mount or Unmount a USB Mass Storage Device With vold Running

1. **Display device aliases for all removable mass storage devices, including USB mass storage devices.**

```
$ eject -n
.
.
.
rmdisk0 -> /vol/dev/rdsk/c4t0d0/clik40    (Generic USB storage)
cdrom0 -> /vol/dev/rdsk/c0t6d0/audio_cd   (Generic CD device)
zip1 -> /vol/dev/rdsk/c2t0d0/fat32        (USB Zip device)
zip0 -> /vol/dev/rdsk/c1t0d0/zip100       (USB Zip device)
```

(continued)

(Continuation)

```
jaz0 -> /vol/dev/rdsk/c3t0d0/jaz1gb       (USB Jaz device)
```

2. **Mount a USB mass storage device by using the device aliases listed previously.**

   ```
   $ volrmmount -i device-alias
   ```

   This example mounts a USB Jaz drive under `/rmdisk/jaz0`.

   ```
   $ volrmmount -i jaz0
   ```

3. **Unmount a USB mass storage device.**

   ```
   $ volrmmount -e device-alias
   ```

   This example unmounts a USB Zip drive from `/rmdisk/zip0`.

   ```
   $ volrmmount -e zip0
   ```

4. **Eject a USB device from a generic USB drive.**

   ```
   $ eject device-alias
   ```

   For example:

   ```
   $ eject rmdisk0
   ```

   Note - The `eject` command also unmounts the device, if it is not unmounted already. The command also terminates any active applications that access the device.

How to Remove a Hot-Pluggable USB Mass Storage Device With vold Running

The following procedure uses a Zip drive as an example of removing a hot-pluggable USB device with vold running.

1. **Unmount the device.**

   ```
   $ volrmmount -e zip0
   ```

2. **(Optional) Stop any active applications that are using the device.**

3. **Eject the device.**

   ```
   $ eject zip0
   ```

4. **Become superuser and stop** vold**.**

   ```
   # /etc/init.d/volmgt stop
   ```

5. **Remove the USB mass storage device.**

6. **Start** vold**.**

   ```
   # /etc/init.d/volmgt start
   ```

How to Add a Hot-Pluggable USB Mass Storage Device With vold Running

This procedure describes how to add a hot-pluggable USB device with vold running.

1. **Insert the USB mass storage device.**

2. **Restart** vold**.**

   ```
   # pkill -HUP vold
   ```

3. **Verify the device has been added.**

   ```
   $ ls device-alias
   ```

Managing USB Mass Storage Devices Without `vold` Running

You can use USB mass storage devices without the volume manager (`vold`) running. Here are two ways to avoid using the volume manager.

1. Stop `vold` by issuing this command.

   ```
   # /etc/init.d/volmgt stop
   ```

2. Keep `vold` running, but do not register the USB mass storage devices with it. Remove volume manager registration of USB mass storage devices by commenting the following line in the `/etc/vold.conf` file, like this:

   ```
   # use rmdisk drive /dev/rdsk/c*s2 dev_rmdisk.so rmdisk%d
   ```

   Once this line is commented, restart `vold`.

   ```
   # pkill -HUP vold_pid
   ```

Caution - If you comment out this line and there are other SCSI or ATAPI Zip or Jaz removable devices in the system, `vold` registration for these devices would be disabled as well.

See `vold.conf(4)` for details.

The following procedures describe how to manage USB mass storage devices without `vold(1M)` running. The device nodes are created under the `/dev/rdsk` directory for character devices and under the `/dev/dsk` directory for block devices. See `scsa2usb(7D)` for details.

How to Mount or Unmount a USB Mass Storage Device Without `vold` Running

1. **Become superuser.**

2. **Mount a USB mass storage device.**

   ```
   # mount -F fs-type /dev/rdsk/cntndnsn /mount-point
   ```

3. **Unmount a USB mass storage device.**

   ```
   # umount /mount-point
   ```

4. **Eject the device.**

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

How to Remove a Hot-Pluggable USB Mass Storage Device Without vold Running

This procedure describes how to remove a hot-pluggable USB device without `vold` running.

1. **Become superuser.**

2. **Remove the hot-pluggable USB device.**

   a. **Unmount the device.**

   ```
   # umount /mount-point
   ```

   b. **(Optional) Stop any active applications that are using the device.**

   c. **Remove the device.**

How to Add a Hot-Pluggable USB Mass Storage Device Without vold Running

This procedure describes how to add a hot-pluggable USB device without `vold` running.

1. **Add a hot-pluggable USB device into the USB port.**

2. **Verify the USB device has been added.**

```
$ ls /dev/rdsk/cntndnsn
```

SPARC Only: USB Power Management

If the system has enabled power management, the USB framework makes a best effort to power-manage all devices. Power-managing a USB device involves the hub driver to suspend the port to which the device is connected. The device might or might not support remote wakeup. If the device supports remote wakeup, it wakes up the hub it is connected to, depending on the event, such as moving the mouse. The host system could also wake the device if an application sends an I/O to it.

All HID (keyboard, mouse, and so forth), hub, and storage devices are power-managed by default if they support the remote wake up capability. A USB printer is power-managed only between two print jobs.

When you power-manage to reduce power consumption, USB leaf devices are powered down first, and after some delay, the parent hub is powered down. When all devices that are connected to this hub's ports are powered down, the hub is powered down after some delay. Avoid cascading many hubs to achieve the most efficient power management.

Hot-Plugging USB Devices

When you plug in a USB device, the device is immediately seen in the system's device hierarchy as displayed in the `prtconf(1M)` command output. When you remove a USB device, the device is removed from the system's device hierarchy, unless the device is in use.

If the USB device is in use when it is removed, the hot-plug behavior is a little different. If a device is in use when it is unplugged, the device node remains, but the driver controlling this device stops all activity on the device. Any new I/O activity issued to this device is returned with an error.

In this situation, the system prompts you to plug in the original device. To recover from accidentally removing a busy USB device, do the following:

1. Plug the original device into the same port.
2. Stop the application that is using the device.
3. Remove the device.

The USB port remains unusable until the original device has been plugged in again. If the device is no longer available, the port remains unusable until the next reboot.

Note - Data integrity might be impaired if you remove an active or open device. Always close the device before removing, except the console keyboard and mouse, which can be moved while active.

USB Cables

Never use USB cable extenders that are available in the market. Always use a hub with longer cables to connect devices. Always use fully rated (12 Mbit/sec) 20/28 AWG cables for connecting USB devices.

SPARC: USB Printer Support

You can now use Solaris Print Manager to set up a USB printer that is attached to a SPARC system with USB ports, starting with the Solaris 8 10/00 release.

The new logical device names for USB printers are:

```
/dev/printers/[1-9]*
```

Therefore, when you add a USB printer to a printer server, select one of these devices for a USB printer under Printer Port on the Add New Attached Printer screen. See *System Administration Guide, Volume 2* for more information on using Solaris Print Manager to set up printers.

Although the new Solaris USB printer driver supports all USB printer-class compliant printers, a list of recommended PostScript™ printers is in the man page, `usbprn(7D)`.

The `usbprn` driver is compliant with non-PostScript printers that utilize third-party PostScript conversion packages like GhostScript. You can obtain conversion packages from the Solaris 8 Software Companion CD, available at `http://www.sun.com/software/solaris/binaries/package.html`.

Refer to the Notes and Diagnostics sections of the `usbprn(7D)` man page for information and cautions about hot-plugging USB printers.

CHAPTER **8**

Managing System Resources

The Solaris 8 software release includes new ways to improve system resources.

Note - For the most current man pages, use the `man` command. The Solaris 8 Update release man pages include new feature information that is not in the *Solaris 8 Reference Manual Collection*.

Extended Accounting Features

The Solaris accounting software has been updated in the Solaris 8 6/00 release. This information supplements information on using system accounting for managing resources as described in "Managing System Accounting (Tasks)" in *System Administration Guide, Volume 2*.

Extended accounting introduces a new variable-length, general-purpose accounting file format that represents general groups of accounting data. Also included is the ability to configure resource utilization that is recorded by the kernel in the various accounting files. Extended accounting features include:

- *Tasks* – New process collectives for tracking resource usage.
- *Projects* – New administrative databases for charging resource usage. Resource usage by a task can be charged to a project.
- `acctadm` – A new tool for configuring various attributes of the extended accounting facility. For example, you can configure the resources that are tracked by the accounting system on a system-wide basis.

The new default accounting configuration requires no administration and causes no complications. If you do use the extended accounting features, however, do not remove the `/etc/project` file, which contains important information about the extended accounting configuration.

Use the following table to find more information about the extended accounting features in this release.

For Information On ...	See ...
Stopping and starting extended accounting	`acctadm(1M)`
Description of the projects database	`projects(4)`
Directly collecting extended accounting data	`libexacct(3LIB)`, `getacct(2)`, `putacct(2)`, and `wracct(2)`

CHAPTER **9**

Managing System Performance

The Solaris 8 software release includes new ways to improve system performance. See the following sections for more information.

- "DNLC Improvements" on page 77
- "*Solaris Tunable Parameters Reference Manual*" on page 79

Note - For the most current man pages, use the `man` command. The Solaris 8 Update release man pages include new feature information that is not in the *Solaris 8 Reference Manual Collection*.

DNLC Improvements

This functionality is updated in the Solaris 8 6/00 software release.

The directory name look-up cache (DNLC) is enhanced in the Solaris 8 6/00 software release to provide improved performance when you access files in large directories with 1000 or more files. The information here adds to information on managing system performance in "System Performance (Overview)" in *System Administration Guide, Volume 2*.

The DNLC is a general file-system service that caches the most recently referenced directory names and their associated vnodes. UFS directory entries are stored linearly on disk. This means that locating an entry requires searching each entry for the name. Adding a new entry requires searching the entire directory to ensure the name does not exist. To solve this performance problem, entire directories are cached in memory by the DNLC.

Another feature in this release is DNLC caching of file objects that have been looked up, but do not exist. This is known as *negative caching*, and is useful because some applications repeatedly test to check if a file exists.

The section that follows describes the new DNLC tunable parameters. These parameters are set optimally and should not be changed casually.

Note - MAXUINT is the maximum value of an unsigned integer.

`dnlc_dir_enable`

Description Enables large directory caching

Data Type Unsigned integer

Default Value 1 (enabled)

Range 0 (disabled), 1 (enabled)

When to Change Directory caching has no known problems, but if problems occur, set `dnlc_dir_enable` to 0 to disable caching.

`dnlc_dir_min_size`

Description Minimum number of entries cached for one directory.

Data Type Unsigned integer.

Default Value 40

Range 0 to MAXUINT (no maximum)

When to Change If performance problems occur with caching small directories, increase `dnlc_dir_min_size`. Note that individual file systems might have their own range limits for caching directories. For instance, UFS limits directories to a minimum of `ufs_min_dir_cache` bytes (approximately 1024 entries), assuming 16 bytes per entry.

`dnlc_dir_max_size`

Description Maximum number of directory entries before caching.

Data Type Unsigned integer.

Default Value MAXUINT (no maximum)

Range 0 to MAXUINT

When to Change If performance problems occur with large directories, decrease `dnlc_dir_max_size`.

Solaris Tunable Parameters Reference Manual

Solaris Tunable Parameters Reference Manual has been updated for the 1/01 release. Information on the semsys:seminfo_semmnu parameter has been added to this book.

To view the book, see *Solaris Tunable Parameters Reference Manual*.