Solaris 8 System Administration Supplement

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

Part Number 806-5183–10
October 2000

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.

Contents

Preface

The *Solaris 8 System Administration Supplement* describes new or changed 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 up-to-date 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 in . . .**
Volume Management Enhancements	
Removable media management now fully supports removable media such as DVD-ROM, Zip and Jaz drives, CD-ROMs, and diskettes. For information on how to use this feature, see Chapter 3.	6/00 Updated in 10/00
System Performance Enhancement	
The Enhanced Directory Name Look-up Cache (DNLC) provides improved performance when accessing files in large directories. For information on how to use this feature, see "DNLC Improvements" on page 39.	6/00
Resource Management Enhancements	
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 recorded by the kernel in the various accounting files. For information on how to use this feature, see "Extended Accounting Features" on page 41.	6/00
Security	

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

Description	First Released in . . .
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. It 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 using the GSS-API can be highly portable. For more information, see *GSS-API Programming Guide*.	6/00
Networking	
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
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 this release, additions include a description of the system properties the CIM Object Manager uses and descriptions of the new Solaris_Printer and other printing definition classes. To see changes to the WBEM book, see “Solaris WBEM Services Administrator’s Guide” on page 43.	10/00
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
USB Support	
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 25. For an overview of USB, see “USB Overview” on page 13.	10/00

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

Description	First Released in . . .
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.	10/00

CHAPTER **2**

USB Support

Note - For the most up-to-date 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*.

USB Overview

This overview is new in the Solaris 8 10/00 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, mice, 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, together. 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 24 for more information.
- Supports a maximum of 126 devices in the Solaris environment.
- Supports a maximum of 12 Mbits per second data transfer.
- Supports low speed (1.5 Mbits per second) and full speed (12 Mbits per second) devices.
- The bus can be easily extended by adding low cost external hubs. Hubs can be connected to hubs forming 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 running the Solaris 8 Intel Platform Edition provide USB support for keyboard and mice 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 supported in the Solaris environment.

These USB Devices ...	Are Supported on These Systems ...
Keyboards and mice	SPARC systems with Sun USB support 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 used in the Solaris environment. Go to `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 is royalty-free and openly available. The specification defines the electrical and mechanical interfaces of the bus and the connectors.

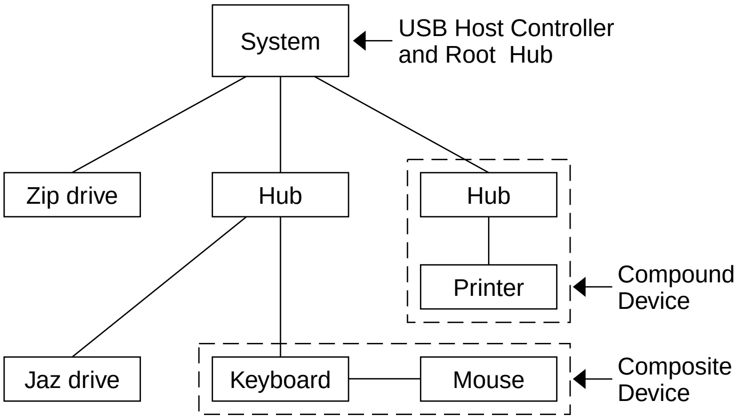

Figure 2–1 USB Physical Device Hierarchy

USB employs a topology where 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 18 for more information about hubs.

The example above shows a system with three active USB ports. The first USB port has a Zip drive that does not have an embedded hub, so additional devices cannot be attached. 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 example above 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 together.

The Human Interface Device (HID) class contains devices that are user controlled such as keyboards, mice, 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. Additional class information can be obtained from the `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 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.

The Figure 2–1 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 SCSA (Sun Common SCSI Architecture). 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 things you should know about USB in the Solaris environment.

USB Keyboards and Mice

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

- The console keyboard and mouse should not be connected through multiple hubs. If they are, the keyboard and mouse might not be enumerated in time, and consequently, will not be configured as the console keyboard and mouse.
- The console keyboard and mouse can be removed at any time *after* a system reboot, and moved to another root hub port or external hub. You cannot move the console keyboard and mouse *during* a reboot or at the ok prompt. After plugging in the keyboard, the keyboard and mouse 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/SHUTDOWN the system using this 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 mice are not supported:
    - The mice enumerate and are usable, but they are not plumbed as console mice.
    - The first mouse that is probed at boot time becomes the console mouse. This might cause confusion if you have multiple mice 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 (3rd party) composite keyboard with a PS/2 mouse, and it is the first one to be probed, it will become 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 mice are supported. A wheel-on-wheel mouse acts like a plain-button mouse. A mouse with more than three buttons function like a 3–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 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 4 levels on either SPARC or IA systems. On SPARC systems, the Open Boot PROM (OBP) cannot reliably probe beyond 4 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

The Solaris 8 10/00 release supports USB Zip, Jaz, Clik, SmartMedia, CompactFlash, and ORB removable mass storage devices. 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 have /usr/dt/man included 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 table below identifies the commands Removable Media Manager uses to manage storage devices from the CDE environment.

Removable Media Manager Uses This Command ...	To ...
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 procedures below 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 procedures below also describe how 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 using the device aliases listed above.**

   ```
   $ 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. It also terminates any active applications accessing the device.

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

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

1. **Unmount the device.**

```
$ volrmmount -e zip0
```

2. **(Optional) Stop any active applications, if any, 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

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

1. **Stop** vold**.**

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

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

3. **Start** vold**.**

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

4. **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 volume manager (`vold`) running. Here are two ways to avoid using volume manager.

- Stop `vold` by issuing this command.

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

- Keep `vold` running, but do not register the USB mass storage devices with it. This can be done by editing `/etc/vold.conf` and commenting the following line, like this:

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

  Once this line is commented, restart `vold(1M)`.

  ```
  # pkill -HUP vold_pid
  ```

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

Managing USB Mass Storage Devices Without `vold` Running

Below are procedures for managing 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

The procedure below 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 using the device.**

   c. **Remove the device.**

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

The procedure below 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 wake up. If the device supports remote wake up, 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 power managing to reduce power consumption, USB leaf devices are powered down first, and after some delay, the parent hub is powered down. When all devices 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. Terminate the application using the device.
3. Remove the device.

The USB port remains unusable until the original device has been plugged back in. 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 Mbits per second) 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 running the Solaris 8 10/00 release.

The new logical device names for USB printers are:

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

Therefore, when adding 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 can be found 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 **3**

Improved Removable Media Management

Note - For the most up-to-date 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*.

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 and Jaz drives, CD-ROMs, and diskettes are mounted and available for reading when 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 found 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 based system to facilitate data transfers to IA based 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 created on a SPARC system is not identical to a UFS file system on PCMCIA or to a diskette 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.

Using CDE’s Removable Media Manager or the volume manager path names such as `/cdrom0`, `/floppy`, `/rmdisk`, `/jaz0`, or `/zip0` are ways to 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 labeled `test` is in diskette drive 0 and a CD 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 using the command-line interface. You can use the volume manager's nickname from the command line by running the `volcheck` command before accessing 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 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 and Zip drives the same way as in previous releases.
- If you are freshly installing the Solaris 8 6/00 release, you cannot access your Jaz and Zip drives 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 formats removable media without certification or with limited certification of certain tracks on the media.
- `long` – This formats removable media completely. For some devices this might include the certification of the whole media by the drive itself.
- `force` – This formats completely without user confirmation. For media with a password protection mechanism, it 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 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 I* 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 I* 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 I* 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 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 or write protection and set a password on Iomega media such as Zip 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 or write protection without a password on Iomega media. In this case, 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 or write protection and a password.**

   a. **Enable read 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 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 **4**

General System Administration

This chapter describes how to use these new system administration features.

- "DNLC Improvements" on page 39
- "Extended Accounting Features" on page 41

Note - For the most up-to-date 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*.

DNLC Improvements

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

The Enhanced Directory Name Look-up Cache (DNLC) is enhanced in the Solaris 8 6/00 software release to provide improved performance when accessing files in large directories with 1000 or more files. The information here adds to information on managing system performance found 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 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, then decrease `dnlc_dir_max_size`.

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 found 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 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 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 **5**

Summary of Changes to Solaris 8 Books

Some Solaris 8 books have been revised and are included in the Solaris 8 10/00 Update Collection. This chapter describes changes to these books since the 6/00 Update release.

Note - For the most up-to-date 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 WBEM Services Administrator's Guide

The following provides an overview of additional information provided in the 10/00 release of the *Solaris WBEM Services Administrator's Guide.*

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.