Solaris 8 System Administration Supplement

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

Part Number 806-3648–10
June 2000

Contents

Preface

The *Solaris 8 System Administration Supplement* describes how to use new system administration features in the Solaris™ 8 6/00 software 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.

Related Books

This document 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 the 6/00 release.

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 operating environment in the 6/00 releases.

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 2.	6/00
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 23.	6/00
Resource Management Enhancements	
Extended accounting introduces a new, variable length, general-purpose accounting file format that represents general groups of accounting data, as well as 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 25.	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 allows 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 Solaris) 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
Internet	
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
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.	6/00

CHAPTER 2

Improved Removable Media Management

Note - For the most up-to-date man pages, use the `man` command. The Solaris 8 6/00 man pages include new feature information not found in the *Solaris 8 Reference Manual Collection*.

Overview

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

Volume management features have been improved in the Solaris 8 6/00software release to fully support removable media. This means DVD-ROMs, Iomega and 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/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 diskette created on an IA system.
- Protect important files on Jaz or Zip drives or diskettes by setting read/write protection. 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*.

▼ 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/
```

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 section above for more information on `rmformat` formatting options.

If the `rmformat` output indicates bad blocks, see the procedure below 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 -H /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
```

(Continuation)

```
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/Write and Password Protection to Removable Media

You can apply read/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 read/write protection without a password.

▼ How to Disable or Enable Read/Write Protection on Removable Media

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

```
$ rmformat -r disable device-name
```

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

b. Enable read or write protection.

```
$ rmformat -r enable device-name
```

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

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

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

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

A password with a maximum of 32 characters can be applied for Iomega media that support this feature. You cannot set read / write protection without a password on Iomega media. In this case, you are prompted to provide a password.

You will 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/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/write protection is enabled or disabled.

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

Examples—Disabling or Enabling Read / 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 3

General System Administration

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

Note - For the most up-to-date man pages, use the `man` command. The Solaris 8 6/00 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 1,000 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 see if a file exists or not.

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, then 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, then 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

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

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, as well as 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`— Anew tool for configuring various attributes of the extended accounting facility. For example, the resources tracked by the accounting system can be configured 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.