Solaris 8 System Administration Supplement

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

Part No: 806–7502–10
April 2001

040210@7940

Contents

Preface 9

1 What's New at a Glance 11

2 Managing With System Administration Tools Topics 17

3 Managing With Solaris Management Console 19
Solaris Management Console Overview 19
Starting Solaris Management Console 21
▼ To Start the Console From the Command Line 21
▼ To Start the SMC Toolbox Editor 21
▼ To Determine if the SMC Server Is Running 21
▼ To Start the SMC Server 22
▼ To Stop the SMC Server 22

4 Managing Desktops, Devices, and Networks With WBEM 23
Changes to the *Solaris WBEM Services Administrator's Guide* 23

5 Managing Security Topics 25

6 Managing Security With RBAC 27
Role-Based Access Control 27

7 Transmission of Data With GSS-API 31
GSS-API Programming Guide 31

8 Managing Security With Smart Cards 33
SPARC: Changes to the *Solaris Smart Cards Administration Guide* 33

9 Managing Servers and Clients Topics 35

10 Managing Diskless Clients 37
Managing Diskless Clients Overview 37
Working With Diskless Client Management 38
User Rights 38
Disk Space Requirements 39
▼ How to Set Up Your Diskless Client Environment 39
Preparing to Add OS Services 40
▼ How to Add an OS Service 42
▼ How to Add a Diskless Client 42
Patching OS Services 43
Troubleshooting 44

11 Managing File Systems Topics 47

12 Managing File Systems With UFS Software 49
Improved UFS Direct I/O Concurrency 49

13 Backing Up File Systems With `fssnap` 51
UFS Snapshots Overview 51
Why Use UFS Snapshots? 52
UFS Snapshots Performance Issues 52
Creating UFS Snapshots 53
▼ How to Create a UFS Snapshot 53
▼ How to Display UFS Snapshot Information 54
Deleting a UFS Snapshot 54
▼ How to Delete a UFS Snapshot 54
Backing Up a UFS Snapshot 55

▼ How to Back Up a UFS Snapshot 56
▼ How to Create an Incremental Dump of a UFS Snapshot 56
Restoring Data From a UFS Snapshot Backup 57

14 Managing Removable Media Topics 59

15 Managing Removable Media 61
Managing Removable Media Overview 61
Accessing Information on Removable Media 62
▼ How to Access Information on Removable Media 63
Accessing Jaz Drives or Zip Drives 63
Formatting Removable Media (`rmformat`) 64
▼ How to Format Removable Media (`rmformat`) 65
▼ How to Format Removable Media for a UFS or UDFS File System 65
▼ How to Format Removable Media for a PCFS File System 66
▼ How to Check a PCFS File System on Removable Media 67
▼ How to Repair Bad Blocks on Removable Media 68
Applying Read or Write and Password Protection to Removable Media 68
▼ How to Enable or Disable Write Protection on Removable Media 68
▼ How to Enable or Disable Read or Write Protection and a Password on Iomega Media 69

16 Managing Devices Topics 71

17 Reconfiguration Coordination Manager (RCM) Scripts 73
RCM Script Overview 73
What Is an RCM Script? 74
What Can an RCM Script Do? 74
How Does the RCM Script Process Work? 74
RCM Script Tasks 75
Application Developer RCM Script Tasks 75
System Administrator RCM Script Tasks 76
Naming an RCM Script 77
Installing or Removing an RCM Script 77
▼ How to Install an RCM Script 77
▼ How to Remove an RCM Script 78

▼ How to Test an RCM Script 78
Tape Backup RCM Script Example 79
What the Tape Backup RCM Script Does 79
Outcomes of the Tape Backup Reconfiguration Scenarios 80
Example—Tape Backup RCM Script 80

18 Managing USB Devices 83
Overview of USB Devices 83
Commonly Used USB Acronyms 84
USB Bus Description 85
About USB in the Solaris Environment 87
USB Keyboards and Mouse Devices 87
USB Host Controller and Root Hub 88
USB Storage Devices 88
Managing USB Mass Storage Devices With `vold` Running 89
▼ How to Mount or Unmount a USB Mass Storage Device With `vold` Running 90
▼ How to Remove a Hot-Pluggable USB Mass Storage Device With `vold` Running 90
▼ How to Add a Hot-Pluggable USB Mass Storage Device With `vold` Running 91
Managing USB Mass Storage Devices Without `vold` Running 91
▼ How to Mount or Unmount a USB Mass Storage Device Without `vold` Running 92
▼ How to Remove a Hot-Pluggable USB Mass Storage Device Without `vold` Running 92
▼ How to Add a Hot-Pluggable USB Mass Storage Device Without `vold` Running 93
SPARC: Creating Data on or Extracting Data From a USB CD 93
▼ How to Prepare for Creating Data on or Extracting Data From a USB CD 93
SPARC Only: USB Power Management 94
Hot-Plugging USB Devices 94
USB Cables 95
USB Printer Support 95

19 Troubleshooting Dynamic Reconfiguration Problems 97
New Dynamic Reconfiguration Error Messages 97

20 Managing Networks Topics 99

21 Mail Services 101

Other Sources of Information About `sendmail` 101

Changes to Version 8.9.3 of `sendmail` 102

New Command-Line Options 102

New and Revised Configuration File Options and Related Topics 103

New Defined Macros for `sendmail` 112

New Macros Used to Build the `sendmail` Configuration File 113

New and Revised `m4` Configuration Macros for `sendmail` and Related Topics 114

New Compile Flags for `sendmail` 117

New Delivery Agent Flags 117

New Equates for Delivery Agents 118

New Queue Features 119

New Uses for LDAP in `sendmail` 119

New Built-in Mailer Feature 120

New Rule Set Features 121

New File Locations 122

Changes to `mail.local` 122

Changes to `mailstats` 123

Changes to `makemap` 123

Other Changes and Features of Interest 124

22 Migration From Berkeley Internet Name Domain (BIND), Version 8.1.2 to BIND Version 8.2.2, Patch Level 5 127

BIND Upgrade 127

23 IP Network Multipathing 129

Detaching Network Adapters 129

24 Mobile IP Administration 131

Reverse Tunneling and Private Addresses 131

25 Managing System Resources Topics 133

26 Managing Resources With System Accounting 135
Extended Accounting Features 135

27 Managing System Performance Topics 137

28 Improving System Performance With DNLC 139
DNLC Improvements 139

29 Managing System Tuning for Better Performance 141
Changes to the *Solaris Tunable Parameters Reference Manual* 141

Preface

The *Solaris 8 System Administration Supplement* describes new features in Solaris™ Update releases. The following information adds to or supersedes information in the previous releases of Solaris 8 documentation sets. Solaris documentation is available on the Solaris 8 Documentation CD.

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℠ 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 that have been 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 With System Administration Tools	

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

Description	First Released
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. 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 19. For information on how to start SMC, see "Starting Solaris Management Console" on page 21. 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 the *Solaris WBEM Services Administrator's Guide*" on page 23 or to view the book, see the *Solaris WBEM Services Administrator's Guide*.
	10/00
	Updated in 1/01

Managing Servers and Clients	

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

Description	First Released
Diskless Client Management is available through the Solaris Management Console command line. You can manage diskless clients, list OS services for diskless clients, and manage patches on all existing diskless clients. For information on diskless client management, see "Managing Diskless Clients Overview" on page 37.	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 27.	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 the *GSS-API Programming Guide*.	6/00
SPARC: The *Solaris Smart Cards Administration Guide* has been updated. In the 1/01 release, 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. In the 4/01 release, previous technical inaccuracies have been corrected. And, new chapters describe the tasks that you need to perform for smart-card setup and additional configuration tasks that you might need to perform if the default smart-card properties are not sufficient for your security environment. To view this book, see the *Solaris Smart Cards Administration Guide*.	1/01 Updated in 4/01
Managing Networks	
Berkeley Internet Name Domain (BIND) version 8.2.2 new functionality includes: ■ `In.named` configuration options – See `conf(4)` man page. ■ Extensions to the resolver (3RESOLV) interface that are safe to use in multithreaded applications. ■ The addition of the `ndc(1M)` command, which is used to start or stop reconfigure `in.named`, and the `dnskeygen (1M)` command, which is used to create TSIG and DNSSEC keys. For more information, see the *Solaris Naming Administration Guide*.	4/01

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

Description	First Released
`sendmail` has new command-line options, new and revised configuration file options, new defined macros, new and revised m4 configuration macros, new and modified compile flags, new delivery agent flags, new equates for delivery agents, new queue features, new uses for LDAP, new rule set features, new file locations, and a new built-in mailer feature. *Mail Services* also describes changes to `mail.local`, changes to `mailstats`, and changes to `makemap`. For more information, see Chapter 21.	4/01
IP network multipathing provides your system with recovery from single-point failures with network adapters and increased traffic throughput. In the 10/00 release, 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. In the 4/01 release, dynamic reconfiguration (DR) uses IP network multipathing to decommission a specific network device without impacting existing IP users. For more information, see the *IP Network Multipathing Administration Guide*.	10/00 Updated in 4/01
Mobile Internet Protocol (IP) enables the transfer of information to and from mobile computers, such as laptop and wireless communications. In the 6/00 release, 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. In the 4/01 release, Mobile IP enables system administrators to set up reverse tunnels. By setting up a reverse tunnel from the mobile node's care-of address to the home agent, you ensure a topologically correct source address for the IP data packet. By using reverse tunnels, system administrators can also assign private addresses to mobile nodes. For more information, see the *Mobile IP Administration Guide*.	6/00 Updated in 4/01
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 the *LDAP Setup and Configuration Guide*.	1/01
Managing File Systems	
Improved UFS Functionality: 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 49.	1/01

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

Description	First Released
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, the documentation 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 53.	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 you create both large and small file systems. However, the biggest `mkfs` performance improvements occur 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 15.	6/00 Updated in 10/00
Managing Devices	
Reconfiguration Coordination Manager (RCM) scripts enable a Solaris system administrator to use a script that cleanly shuts down devices and applications during a dynamic reconfiguration operation. For more information, see Chapter 17.	4/01
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 "USB Printer Support" on page 95. For an overview of USB, see "Overview of USB Devices" on page 83.	10/00 Updated in 1/01 and again in 4/01
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 97.	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 135.	6/00
Managing System Performance	

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

Description	First Released
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 139.	6/00
The *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 the *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 With System Administration Tools Topics

This section provides instructions for managing with system administration tools in the Solaris environment. This section contains these chapters.

Chapter 3	Provides information on SMC tools and instructions for starting and stopping SMC
Chapter 4	Describes changes to the *Solaris WBEM Services Administration Guide*

CHAPTER 3

Managing With Solaris Management Console

The Solaris Management Console is new in the Solaris 8 1/01 release. For general information about Solaris system management, see the *System Administration Guide, Volume 1*.

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*.

Solaris Management Console Overview

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 21. 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`
  - List OS services for diskless clients and manage patches on all existing diskless clients – `smosservice.1m`

  For more information on diskless clients, see Chapter 10.
- 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

- **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

- **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:

- **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

- **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

- **As root, type:**

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

A message should return: "SMC stopped."

CHAPTER 4

Managing Desktops, Devices, and Networks With WBEM

The *Solaris WBEM Services Administrator's Guide* has been updated with the following information for the 10/00 and 1/01 releases.

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*.

Changes to the *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.

TABLE 4–1 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.

TABLE 4–1 Changes to WBEM *(Continued)*

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 5

Managing Security Topics

This section provides instructions for managing security in the Solaris environment. This section contains these chapters.

Chapter 6	Describes changes to RBAC including: a terminology change, changes in rights profiles, and changes in `policy.conf(4)`
Chapter 7	Describes the Generic Security Services Application Programming Interface (GSS-API)
Chapter 8	Summarizes changes to the *Solaris Smart Cards Administration Guide*

CHAPTER 6

Managing Security With RBAC

The role-based access control (RBAC) databases have been enhanced in the Solaris 8 1/01 release. The following information supplements information on the RBAC databases that is in "Role-Based Access Control" in the *System Administration Guide, Volume 2*.

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

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 19. 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.

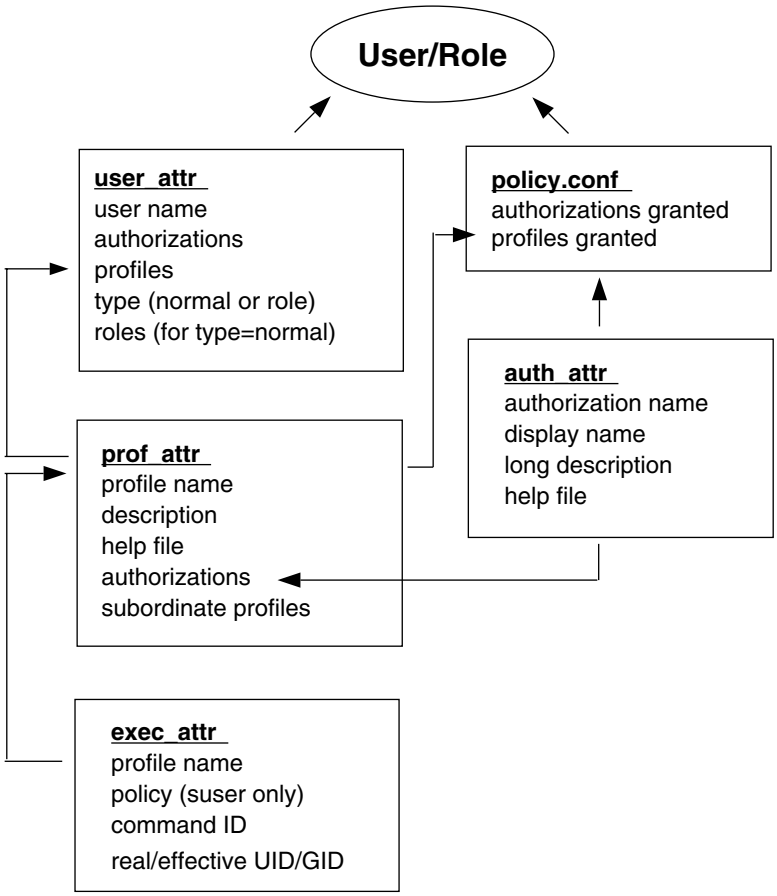

FIGURE 6–1 Extended Attribute Databases

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 6–1 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(1M)`	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
`smmaillist(1M)`	solaris.admin.usermgr.read, solaris.admin.usermgr.write
`smosservice(1M)`	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

CHAPTER 7

Transmission of Data With GSS-API

The *GSS-API Programming Guide* is new in the Solaris 8 6/00 release.

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*.

GSS-API Programming Guide

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 regarding 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 the *GSS-API Programming Guide*.

CHAPTER 8

Managing Security With Smart Cards

The *Solaris Smart Cards Administration Guide* has been updated with the following information for the 1/01 and 4/01 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*.

SPARC: Changes to the *Solaris Smart Cards Administration Guide*

The *Solaris Smart Cards Administration Guide* documents new security functionality in the Solaris 8 software release.

- The chapter on setting up card readers now has information on setting up internal card readers.
- New chapters describe the tasks that you need to perform for smart-card setup and additional configuration tasks that you might need to perform if the default smart-card properties are not sufficient for your security environment. These instructions have been further streamlined in the 4/01 release to make the smart card setup process easier.
- Previous technical inaccuracies have been corrected.

CHAPTER 9

Managing Servers and Clients Topics

This section provides instructions for writing device drivers in the Solaris environment. This section contains this chapter:

Chapter 10	Provides instructions for setting up and managing a diskless client environment

CHAPTER 10

Managing Diskless Clients

Diskless Client Management is new in the Solaris 8 1/01 release and updates the Solstice AdminSuite™ 2.3 Diskless Client tool.

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*.

Managing Diskless Clients Overview

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 or compatible 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 10–1, you can easily set up and manage your diskless client environment.

TABLE 10–1 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, subcommand, and required options. For example, to display the usage statement for `smdiskless add`:

  `%/usr/sadm/bin/smdiskless add -p` *my_password* **`-u`** *my_user_name* **`-- -h`**

- *Man pages* – To view a man page, type `man` and the command name. For example, to display the man page for `smdiskless`:

  `%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 10–2 lists the rights that are required to use the Diskless Client Management commands.

TABLE 10–2 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	All tasks

Disk Space Requirements

Before you set up your diskless client environment, make sure you have the required disk space available for each of the Diskless Client directories.

TABLE 10–3 Disk Space Requirements

Directory	Required Space (MB)
`/export/Solaris_`*version*	10
`/export/exec`	800
`/export/share`	5
`/export/swap/`*diskless_client*	32 (default size)
`/export/dump/`*diskless_client*	32 (default size)
`/export/root/templates/Solaris_`*version*	30
`/export/root/clone/Solaris_`*version*`/`*machine_class*	30 through 60 (depends on machine class)
`/export/root/`*diskless_client* (clone of above)	30 through 60 (depends on machine class)
`/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 `admhostdel` command.**
3. **Remove the existing AdminSuite 2.3™ OS services by using the `admhostmod` command.**
4. **Upgrade the machine(s) designated as the OS server(s) to the Solaris 8 1/01 or compatible operating environment.**
5. **To view Diskless Client error messages by using the SMC Log Viewer tool, start the SMC:**

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

   Then choose Log Viewer from the SMC main window.
6. **Add the required OS services.**
7. **Add the diskless client(s).**
8. **Boot each diskless client from the PROM level by using the `boot net` command. For more information on this command, refer to the** *Solaris 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 the *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` Specify the `SUNWCXall` cluster for each diskless client because the machine that runs `sun4u` requires `SUNWCXall`. In addition, 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/root/templates/Solaris_`*version*
`/export/root/clone`
`/export/root/clone/Solaris_`*version*
`/export/root/clone/Solaris_`*version*`/`*machine_class*

1. **Use the `smosservice add` command, including the required mediapath, platform, and cluster options, to add the first OS service. The installation process can require 45 minutes, depending on the server speed and the OS service configuration you choose.**
2. **Continue to use the `smosservice add` command to add other OS services.**
3. **When you are finished adding OS services, use the `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*
`/export/swap/`*diskless_client*
`/tftpboot/`*diskless_client_ipaddress_in_hex*`/export/dump/`*diskless_client* (if you specify the `-x` *dump* option)

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.

---

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. 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 `smdiskless add` command to add each diskless client.**
3. **When you are finished adding diskless clients, use the `smdiskless list` command to verify that the diskless clients were installed.**

Patching OS Services

You use the `smosservice patch` command to do the following:

- Establish the `/export/diskless/Patches` patch spool directory on an OS server.
- Add patches to the patch spool directory. If the patch being added obsoletes an existing patch in the spool, the obsolete patch is moved to `/export/diskless/Patches/Archive`.
- 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 – Keep your OS servers up to date by installing recommended OS patches on a timely basis.

Displaying Patches

Diskless client patches are logged in different directories, depending on the type of patch:

- Kernel patches are logged in the diskless client's `/var/sadm/patch` directory. To display kernel patches from the diskless client, type:

  ```
  % showrev -p
  ```

- `/usr` patches are logged in the OS server's `/export/Solaris_`*version*`/var/patch` directory. A directory is created for each patch ID. To list the patches, change to this directory and type:

  ```
  % ls -l
  ```

To list all spooled patches by OS and architecture, use the `smosservice` command with the `-P` option.

Troubleshooting

This section lists some common problems with Diskless Client Management and possible solutions.

Problem

- OS server does not respond to client RARP requests
- OS server does not respond to client `bootparam` requests
- OS server cannot mount diskless client root file system

Solution

In a files environment

- Verify that `files` is listed as the first source for `hosts`, `ethers`, and `bootparams` in `/etc/nsswitch.conf` on the OS server.
- Verify that the client's IP address appears in `/etc/inet/hosts`.
- Verify that the client's Ethernet address appears in `/etc/ethers`.
- Verify that the `/etc/bootparams` file contains the following paths to the client's root and swap areas:

  diskless_client `root=`*os_server*`:/export/`
  `root/`*diskless_client* `swap=`*os_server*`:/export/`
  `swap/`*diskless_client* `swapsize=24`

  The swap size varies depending on whether you specify the `-x` *swapsize* option when you add the diskless client. If you specify the `-x` *dump* option when you add the diskless client, the following line is present:

  `dump=`*os_server*`:/export/dump/`*diskless_client* `dumpsize=24`

  The dump size varies depending on whether you specify the `-x` *dumpsize* option when you add the diskless client.

- Verify that the OS server's IP address appears in `/export/root/`*diskless_client*`/etc/inet/hosts`.

In a name service environment

- Verify that both the OS server's and the client's Ethernet address and IP address are correctly mapped.
- Verify that `/etc/bootparams` contains the paths to the client's root and swap areas, as follows:

  diskless_client `root=`*os_server*`:/export/`
  `root/`*diskless_client* `swap=`*os_server*`:/export/`
  `swap/`*diskless_client* `swapsize=24`

  The swap size varies depending on whether you specify the `-x` *swapsize* option when you add the diskless client. If you specify the `-x` *dump* option when you add the diskless client, the following line is present:

  `dump=`*os_server*`:/export/dump/`*diskless_client* `dumpsize=24`

The dump size varies depending on whether you specify the `-x` *dumpsize* option when you add the diskless client.

Problem
Diskless client panics

Solution

- Verify that the OS server's Ethernet address is correctly mapped to its IP address. If you physically moved a machine from one network to another, you might have forgotten to remap the machine's new IP address.
- Verify that the client's host name, IP address, and Ethernet address do not exist in the database of another server *on the same subnet* that responds to the client's RARP, TFTP, or `bootparam` requests. Often, test machines are set up to install their OS from an install server. In these cases, the install server answers the client's RARP or `bootparam` request, returning an incorrect IP address. This incorrect address might result in the download of a boot program for the wrong architecture, or a failure to mount the client's root file system.
- Verify that the diskless client's TFTP requests are not answered by an install server (or previous OS server) that transfers an incorrect boot program. If the boot program is of a different architecture, the client immediately panics. If the boot program loads from a non-OS server, the client might obtain its root partition from the non-OS server and its `/usr` partition from the OS server. In this situation, the client panics if the root and `/usr` partitions are of conflicting architectures or versions.
- If you are using both an install server and an OS server, verify that the following entry exists in `/etc/dfs/dfstab`:

  ```
  share -F nfs -o -ro /export/exec/Solaris_version_instruction_set.all/usr
  ```

  Where *version*=`2.6`, `2.7`, or `8`, and *instruction_set*=`sparc` or `i386`.
- Verify that the diskless client's root, `/swap`, and `/dump` (if specified) partitions have share entries:

  ```
  % share -F nfs -o rw=client_name,root=client_name /export/root/client_name % share -F
  ```
- On the OS server, type the following to check which files are shared:

  ```
  % share
  ```

  The OS server must share `/export/root/`*client_name* and `/export/swap/`*client_name* (defaults), or the root, `/swap`, and `/dump` partitions you specified when you added the diskless client.

  Verify that the following entry exists in `/etc/dfs/dfstab`:

  ```
  % share -F nfs -o ro /export/exec/Solaris_version_instruction_set.all/usr
  ```

Problem
OS server is not responding to diskless client's RARP request

Solution

From the client's intended OS server, run `snoop` as root by using the client's Ethernet address:

```
# snoop xx:xx:xx:xx:xx:xx
```

Problem

Boot program downloads, but panics early in the process

Solution

Using `snoop`, verify that the intended OS server is answering the client's TFTP and NFS requests.

Problem

- Diskless client hangs
- Incorrect server responds to diskless client's RARP request

Solution

Restart the following on the OS server:

```
% /usr/bin/rpc.bootparamd% /usr/sbin/in.rarpd -a
```

CHAPTER 11

Managing File Systems Topics

This section provides instructions for managing file systems in the Solaris environment. This section contains these chapters.

Chapter 12	Describes improvements to UFS direct I/O performance due to concurrent read and write access to regular UFS files.
Chapter 13	Describes how to back up a file system by using the new `fssnap` command to create a read-only snapshot of the system

CHAPTER 12

Managing File Systems With UFS Software

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

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 Solaris 8 software release includes new UFS functionality. 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 updated 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.

CHAPTER 13

Backing Up File Systems With `fssnap`

The `fssnap` command is new in the Solaris 8 1/01 release. The following information supplements information on backing up file systems that is in "Backing Up and Restoring File Systems (Overview)" in the *System Administration Guide, Volume 1*.

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*.

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.

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 bitmapped 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 you used 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.

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 when the file system is written to, but they will see no impact when 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 that the file system has 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 deleted. 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 the *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 the *System Administration Guide, Volume 1*.

CHAPTER 14

Managing Removable Media Topics

This section provides instructions for managing removable media in the Solaris environment. This section contains this chapter.

Chapter 15	Describes improvements in Solaris volume management features. Both the Common Desktop Environment (CDE) volume management and Solaris command-line features are enhanced to fully support removable media.

CHAPTER 15

Managing Removable Media

Managing removable media has been enhanced in the Solaris 8 6/00 release, and the documentation about this feature has been updated again in the 10/00 release. The following information supplements information on managing removable media that is in "Guidelines for Using CDs and Diskettes (Overview)" in the *System Administration Guide, Volume 1*. For information on using File Manager to administer this feature, see "Removable Media" in the *Solaris 8 Desktop User Supplement*.

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

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.

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 With File Manager" in the *Solaris Common Desktop Environment: User's Guide*.

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 in 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, the use of 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 the *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 the *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 the *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 protection 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 16

Managing Devices Topics

This section provides instructions for managing devices in the Solaris environment. This section contains these chapters.

Chapter 17	Describes how you can use the new RCM script feature to write your own scripts to shut down your applications, and to cleanly release devices from your applications during dynamic reconfiguration
"Overview of USB Devices" on page 83	Provides an overview of Universal Serial Bus (USB) devices and how they are supported in the Solaris operating environment
"USB Printer Support" on page 95	Provides instructions on setting up a USB printer by using Solaris Print Manager
Chapter 19	Provides troubleshooting information for the dynamic reconfiguration software

CHAPTER 17

Reconfiguration Coordination Manager (RCM) Scripts

The Reconfiguration Coordination Manager (RCM) scripts feature is new in the Solaris 8 4/01 release. For general information about Solaris system management, see the *System Administration Guide, Volume 1*.

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*.

RCM Script Overview

Dynamic reconfiguration of system resources enables you to reconfigure system components while the system is still running. This feature has been available with the `cfgadm` command since the Solaris 8 release. The Reconfiguration Coordination Manager is the framework that manages the dynamic removal of system components. By using RCM, you can register and release system resources in an orderly manner.

In the Solaris 8 4/01 release, you can use the new RCM script feature to write your own scripts to shut down your applications, or to cleanly release the devices from your applications during dynamic reconfiguration. The RCM framework launches a script automatically in response to a reconfiguration request, if the request impacts the resources that are registered by the script.

Previously, you had to release resources from applications manually before you could dynamically remove the resource. Or, you could use the `cfgadm` command with the `-f` option to force a reconfiguration operation, but this option might leave your applications in an unknown state. Also, the manual release of resources from applications commonly causes errors.

The RCM script feature simplifies and better controls the dynamic reconfiguration process. By creating an RCM script, you can:

- Automatically release a device when you dynamically remove a device. This process also closes the device if the device is opened by an application.
- Run site-specific tasks when you dynamically remove a device from the system.

What Is an RCM Script?

An RCM script is:

- An executable shell script (Perl, `sh`, `csh`, or `ksh`) or binary program that the RCM daemon runs. Perl is the recommended language.
- A script that runs in its own address space by using the user ID of the script file owner.
- A script that is run by the RCM daemon when you use the `cfgadm` command to dynamically reconfigure a system resource.

What Can an RCM Script Do?

You can use an RCM script to release a device from an application when you dynamically remove a device. If the device is currently open, the RCM script also closes the device.

For example, an RCM script for a tape backup application can inform the tape backup application to close the tape drive or shut down the tape backup application.

How Does the RCM Script Process Work?

You can invoke a script as follows:

`$` *`script-name command`* `[`*`args`* `...]`

A script performs the following basic steps:

1. Takes the RCM command from command-line arguments.
2. Executes the command.
3. Writes the results to `stdout` as name-value pairs.
4. Exits with the appropriate exit status.

The RCM daemon runs one instance of a script at a time. For example, if a script is running, the RCM daemon does not run the same script until the first script exits.

RCM Script Commands

You must include the following RCM commands in an RCM script:

- `scriptinfo` – Gathers script information
- `register` – Registers interest in resources
- `resourceinfo` – Gathers resource information

You might include some or all of the following RCM commands:

- `queryremove` – Queries whether the resource can be released
- `preremove` – Releases the resource
- `postremove` – Provides post-resource removal notification
- `undoremove` – Undoes the actions done in `preremove`

See the `rcmscript`(4) man page for a complete description of these RCM commands.

RCM Script Processing Environment

When you dynamically remove a device, the RCM daemon runs:

- The script's `register` command to gather the list of resources (device names) that are identified in the script.
- The script's `queryremove/preremove` commands prior to removing the resource if the script's registered resources are affected by the dynamic remove operation.
- The script's `postremove` command if the remove operation succeeds. However, if the remove operation fails, the RCM daemon runs the script's `undoremove` command.

RCM Script Tasks

The following sections describe the RCM script tasks for application developers and system administrators.

Application Developer RCM Script Tasks

The following table describes the tasks for an application developer who is creating an RCM script.

TABLE 17–1 Application Developer RCM Script Task Map

Task	Description	For Instructions, Go To
1. Identify Resources Your Application Uses	Identify the resources (device names) your application uses that you could potentially dynamically remove.	`cfgadm`(1m) man page
2. Identify Commands to Release the Resource	Identify the commands for notifying the application to cleanly release the resource from the application.	Application documentation
3. Identify Commands for Post-Removal of the Resource	Include the commands for notifying the application of the resource removal.	`rcmscript`(4) man page
4. Identify Commands If the Resource Removal Fails	Include the commands for notifying the application of the available resource.	`rcmscript`(4) man page
5. Write the RCM Script		"Tape Backup RCM Script Example" on page 79
6. Install the RCM Script	Add the script to the appropriate script directory.	"How to Install an RCM Script" on page 77
7. Test the RCM Script	Test the script by running the script commands manually and by initiating a dynamic reconfiguration operation.	"How to Test an RCM Script" on page 78

System Administrator RCM Script Tasks

The following table describes the tasks for a system administrator who is creating an RCM script to do site customization.

TABLE 17–2 System Administrator RCM Script Task Map

Task	Description	For Instructions, Go To
1. Identify Resources to Be Dynamically Removed	Identify the resources (device names) to be potentially removed by using the `cfgadm -l` command.	`cfgadm`(1m) man page
2. Identify Applications to Be Stopped	Identify the commands for stopping the applications cleanly.	Application documentation
3. Identify Commands For Pre- and Post-Removal of the Resource	Identify the actions to be taken before and after the resource is removed.	`rcmscript`(4) man page
4. Write the RCM Script		"Tape Backup RCM Script Example" on page 79

TABLE 17–2 System Administrator RCM Script Task Map *(Continued)*

Task	Description	For Instructions, Go To
5. Install the RCM Script	Add the script to the appropriate script directory.	"How to Install an RCM Script" on page 77
6. Test the RCM Script	Test the script by running the script commands manually and by initiating a dynamic reconfiguration operation.	"How to Test an RCM Script" on page 78

Naming an RCM Script

A script must be named as *vendor,service* where the following applies:

vendor	Is the stock symbol of the vendor that provides the script, or any distinct name that identifies the vendor.
service	Is the name of the service that the script represents.

Installing or Removing an RCM Script

You must be superuser (root) to install or remove an RCM script. Use this table to determine where you should install your RCM script.

TABLE 17–3 RCM Script Directories

Directory Location	Script Type
`/etc/rcm/scripts`	Scripts for specific systems
`/usr/platform/‘uname -i‘/lib/rcm/scripts`	Scripts for a specific hardware implementation
`/usr/platform/‘uname -m‘/lib/rcm/scripts`	Scripts for a specific hardware class
`/usr/lib/rcm/scripts`	Scripts for any hardware

▼ How to Install an RCM Script

1. **Become superuser.**
2. **Copy the script to the appropriate directory as described in Table 17–3.**
   For example:

   ```
   # cp SUNW,sample.pl /usr/lib/rcm/scripts
   ```

3. **Change the user ID and the group ID of the script to the desired values.**

   For example:

   ```
   # chown user:group /usr/lib/rcm/scripts/SUNW,sample.pl
   ```

4. **Send `SIGHUP` to the RCM daemon.**

   ```
   # pkill -HUP -x -u root rcm_daemon
   ```

▼ How to Remove an RCM Script

1. **Become superuser.**
2. **Remove the script from the RCM script directory.**

   For example:

   ```
   # rm /usr/lib/rcm/scripts/SUNW,sample.pl
   ```

3. **Send `SIGHUP` to the RCM daemon.**

   ```
   # pkill -HUP -x -u root rcm_daemon
   ```

▼ How to Test an RCM Script

1. **Set environment variables, such as `RCM_ENV_FORCE`, on the command-line shell before running your script.**

   For example, in the Korn shell, use:

   ```
   $ export RCM_ENV_FORCE=TRUE
   ```

2. **Test the script by running the script commands manually from the command line.**

   For example:

   ```
   $ script-name scriptinfo
   $ script-name register
   $ script-name preremove resource-name
   $ script-name postremove resource-name
   ```

3. **Make sure each RCM script command in your script prints appropriate output to `stdout`.**
4. **Install the script in the appropriate script directory.**

   See "How to Install an RCM Script" on page 77 for more information.

5. **Test the script by initiating a dynamic remove operation:**

   For example, assume your script registers the device, `/dev/dsk/c1t0d0s0`. Try these commands.

```
$ cfgadm -c unconfigure c1::dsk/c1t0d0
$ cfgadm -f -c unconfigure c1::dsk/c1t0d0
$ cfgadm -c configure c1::dsk/c1t0d0
```

Caution – Make sure you are familiar with these commands because they can alter the state of the system and can cause system failures.

Tape Backup RCM Script Example

This example illustrates how to use an RCM script for tape backups.

What the Tape Backup RCM Script Does

The tape backup RCM script performs the following steps:

1. Sets up a dispatch table of RCM commands.
2. Calls the dispatch routine that corresponds to the specified RCM command and exits with status 2 for unimplemented RCM commands.
3. Sets up the `scriptinfo` section:

   ```
   rcm_script_func_info=Tape backup appl script for DR
   ```

4. Registers all tape drives in the system by printing all tape drive device names to `stdout`.

   ```
   rcm_resource_name=/dev/rmt/$f
   ```

   If an error occurs, prints the error information to `stdout`.

   ```
   rcm_failure_reason=$errmsg
   ```

5. Sets up the resource information for the tape device.

   ```
   rcm_resource_usage_info=Backup Tape Unit Number $unit
   ```

6. Sets up the `preremove` information by checking if the backup application is using the device. If the backup application is not using the device, the dynamic reconfiguration operation continues. If the backup application is using the device, the script checks `RCM_ENV_FORCE`. If `RCM_ENV_FORCE` is set to `FALSE`, the script denies the dynamic reconfiguration operation and prints the following message:

   ```
   rcm_failure_reason=tape backup in progress pid=...
   ```

   If `RCM_ENV_FORCE` is set to `TRUE`, the backup application is stopped, and the reconfiguration operation proceeds.

Outcomes of the Tape Backup Reconfiguration Scenarios

Here are the various outcomes if you use the `cfgadm` command to remove a tape device without the RCM script.

- If you use the `cfgadm` command and the backup application is not using the tape device, the operation succeeds.
- If you use the `cfgadm` command and the backup application is using the tape device, the operation fails.

Here are the various outcomes if you use the `cfgadm` command to remove a tape device with the RCM script.

- If you use the `cfgadm` command and the backup application is not using the tape device, the operation succeeds.
- If you use the `cfgadm` command without the `-f` option and the backup application is using the tape device, the operation fails with an error message similar to the following:

  ```
  tape backup in progress pid=...
  ```

- If you use the `cfgadm -f` command and the backup application is using the tape device, the script stops the backup application and the `cfgadm` operation succeeds.

Example—Tape Backup RCM Script

```
#! /usr/bin/perl -w
   #
   # A sample site customization RCM script.
   #
   # When RCM_ENV_FORCE is FALSE this script indicates to RCM that it cannot
   # release the tape drive when the tape drive is being used for backup.
   #
   # When RCM_ENV_FORCE is TRUE this script allows DR removing a tape drive
   # when the tape drive is being used for backup by killing the tape
   # backup application.
   #

    use strict;

    my ($cmd, %dispatch);
    $cmd = shift(@ARGV);
# dispatch table for RCM commands
    %dispatch = (
            "scriptinfo"    =>      \&do_scriptinfo,
            "register"      =>      \&do_register,
            "resourceinfo"  =>      \&do_resourceinfo,
            "queryremove"   =>      \&do_preremove,
            "preremove"     =>      \&do_preremove
```

```
    );


    if (defined($dispatch{$cmd})) {
            &{$dispatch{$cmd}};
    } else {
            exit (2);
    }

    sub do_scriptinfo
    {
            print "rcm_script_version=1\n";
            print "rcm_script_func_info=Tape backup appl script for DR\n";
            exit (0);
    }

    sub do_register
{
            my ($dir, $f, $errmsg);

            $dir = opendir(RMT, "/dev/rmt");
            if (!$dir) {
                 $errmsg = "Unable to open /dev/rmt directory: $!";
                 print "rcm_failure_reason=$errmsg\n";
                 exit (1);
            }

            while ($f = readdir(RMT)) {
                # ignore hidden files and multiple names for the same device
                if (($f !~ /^\./) && ($f =~ /^[0-9]+$/)) {
                        print "rcm_resource_name=/dev/rmt/$f\n";
                    }

            }

            closedir(RMT);
            exit (0);
    }
sub do_resourceinfo
    {
        my ($rsrc, $unit);

        $rsrc = shift(@ARGV);
        if ($rsrc =~ /^\/dev\/rmt\/([0-9]+)$/) {
             $unit = $1;
             print "rcm_resource_usage_info=Backup Tape Unit Number $unit\n";
             exit (0);
        } else {
             print "rcm_failure_reason=Unknown tape device!\n";
             exit (1);
        }
    }

    sub do_preremove
    {
```

```
                my ($rsrc);

                $rsrc = shift(@ARGV);

                # check if backup application is using this resource
                #if (the backup application is not running on $rsrc) {
                        # allow the DR to continue
                #        exit (0);
                #}
                #
                # If RCM_ENV_FORCE is FALSE deny the operation.
                # If RCM_ENV_FORCE is TRUE kill the backup application in order
                # to allow the DR operation to proceed
                #
                if ($ENV{RCM_ENV_FORCE} eq 'TRUE') {
                     if ($cmd eq 'preremove') {
                             # kill the tape backup application
                     }
                     exit (0);
                } else {
                     #
                     # indicate that the tape drive can not be released
                     # since the device is being used for backup by the
                     # tape backup application
                     #
                     print "rcm_failure_reason=tape backup in progress pid=...\n"
;
                     exit (3);
                }
    }
```

CHAPTER 18

Managing USB Devices

This chapter on managing USB devices has been revised in the Solaris 8 4/01 software release. See the following sections for further information.

- "Overview of USB Devices" on page 83
- "USB Printer Support" on page 95

For general information about device management in Solaris, see "Managing Devices Topics" in the *System Administration Guide, Volume 1*.

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*.

Overview of USB Devices

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 94 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 that run 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. 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 and IA.
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)

Acronym	Definition
USBAI	USBA Client Driver Interface (Solaris)
HCD	USB host controller driver

USB Bus Description

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

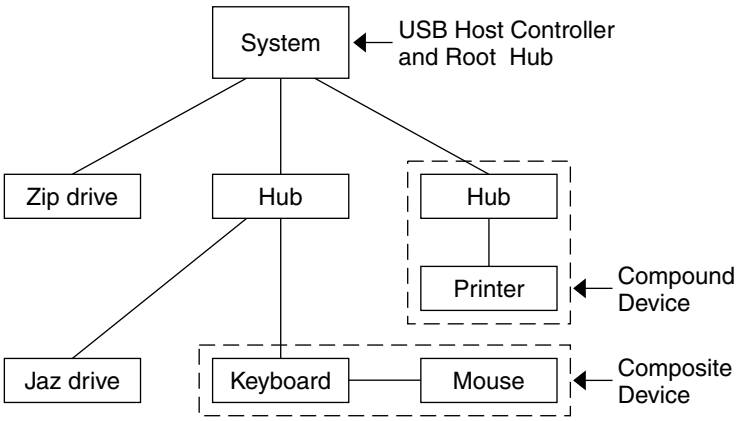

FIGURE 18–1 USB Physical Device Hierarchy

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 88 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 that are 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 in 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 *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 by using compatible names properties, as defined in section 3.3.2.1 of the 1275 binding.

Figure 18–1 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 that are connected anywhere on the bus are configured as console keyboard and mouse. Booting the system is slower 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 you plug 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 is probed at boot time becomes the console keyboard. The result of this probing might cause confusion if multiple keyboards are 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. The result of this probing 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 cannot work because it is not configured as the console mouse.
- Only two-button 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 derives 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 that are 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)
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.

- 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. 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
```

  After this line is commented, restart `vold`.

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

Caution – If you comment out this line and other SCSI or ATAPI Zip or Jaz removable devices are 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/dsk/c***n***t***n***d***n***s***n* */mount-point*

   This command might fail it the device is read only. Use the following command for CD-ROM devices.

   # **mount -F** *fs-type* **-o ro /dev/dsk/c***n***t***n***d***n***s***n* */mount-point*

   For example:

   ```
   # mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /mnt
   ```

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

   # **umount** */mount-point*

4. **Eject the device.**

   # **eject /dev/**[**r**]**dsk/c***n***t***n***d***n***s***n*

▼ 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/c***n***t***n***d***n***s***n***

SPARC: Creating Data on or Extracting Data From a USB CD

You can use the `cdrw` command to create and extract data from audio CDs. The `cdrw` command is available on the Software Supplement for the Solaris 8 Operating Environment 1/01 CD.

- SCSI, ATAPI, and USB CD devices are supported. Currently, the only CD-RW device supported by Sun is the Sony Spress USB CD-RW.
- The CD-R or CD-RW drive must be MMC compliant.

See the `cdrw` man page in the *Solaris on Sun Hardware Reference Manual Supplement* for information on using this command.

▼ How to Prepare for Creating Data on or Extracting Data From a USB CD

The `cdrw` command works with or without `vold` running. See the `cdrw`(1) and `mkisofs`(1M) man pages for more information.

1. **Insert a CD into the CD-RW device.**

The CD can be any CD that the device can read.

2. **Check that the CD-RW drive is connected properly by listing the device.**

```
# cdrw -l
    Node               |    Connected Device             |  Device type
-----------------------+---------------------------------+-----------------
 /dev/rdsk/c0t0d0s2    | SONY     CD-RW  CRX120E   1.0k | CD Reader/Writer
```

3. **(Optional) If you do not see the drive in the list, you might have to do a reconfiguration boot so that the system recognizes the device.**

```
# touch /reconfigure
# init 6
```

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 means the hub driver suspends 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 wakeup 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. To achieve the most efficient power management, do not cascade many hubs.

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.

USB Printer Support

You can 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. You can also set up USB printers on IA systems, starting with the Solaris 8 04/01 release.

The new logical device names for USB printers are:

`/dev/printers/[0...N]*`

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 the *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 `usbprn(7D)` man page.

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 19

Troubleshooting Dynamic Reconfiguration Problems

The dynamic reconfiguration software has been enhanced in the Solaris 8 1/01 release. The following information supplements information on troubleshooting dynamic reconfiguration problems that is in "Configuring Devices" in the *System Administration Guide, Volume 1*.

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 to improve troubleshooting dynamic reconfiguration problems.

If you use the `cfgadm` command to remove a system resource, such as a swap device or a dedicated dump device, error messages are 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.

CHAPTER 20

Managing Networks Topics

This section provides instructions for managing networks in the Solaris environment. This section contains these chapters.

Chapter 21	Describes significant changes made to version 8.9.3 of `sendmail`, the version that was included in the Solaris™ 8 release
"BIND Upgrade" on page 127	Summarizes new functionality in Berkeley Internet Name Domain (BIND) version 8.2.2 in the Solaris 8 4/01 release
Chapter 23	Summarizes changes to IP network multipathing that enable dynamic reconfiguration to decommission a specific network device without impacting existing IP users
Chapter 24	Summarizes changes to Mobile IP that enable system administrators to set up reverse tunnels and to assign private addresses to mobile nodes

CHAPTER 21

Mail Services

Significant changes have been made to version 8.9.3 of `sendmail`, the version that was included in the Solaris™ 8 release. This chapter documents those changes, which have been incorporated into version 8.10.2+Sun of `sendmail`, the new version in this Solaris 8 4/01 release. The following lists the major sections in this chapter.

- "Other Sources of Information About `sendmail`" on page 101
- "Changes to Version 8.9.3 of `sendmail`" on page 102
- "Changes to `mail.local`" on page 122
- "Changes to `mailstats`" on page 123
- "Changes to `makemap`" on page 123
- "Other Changes and Features of Interest" on page 124

For information about Solaris Mail Services, see "Mail Services Topics" in the *System Administration Guide, Volume 3*.

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*.

Other Sources of Information About `sendmail`

The following list provides additional sources for information about `sendmail`.

- Home page for `sendmail` – http://www.sendmail.org
- FAQ for `sendmail` – http://www.sendmail.org/faq

- README for new `sendmail` configuration files – http://www.sendmail.org/m4/readme.html
- Fatbrain.com for books about `sendmail`, particularly the second edition of *sendmail* from O'Reilly & Associates, Inc. – http://www1.fatbrain.com/catalogs/computing/subjects.asp?SubjectCode=OML

Changes to Version 8.9.3 of `sendmail`

This section contains information on the following topics.

- "New Command-Line Options" on page 102
- "New and Revised Configuration File Options and Related Topics" on page 103
- "New Defined Macros for `sendmail`" on page 112
- "New and Revised `m4` Configuration Macros for `sendmail` and Related Topics" on page 114
- "New Compile Flags for `sendmail`" on page 117
- "New Delivery Agent Flags" on page 117
- "New Equates for Delivery Agents" on page 118
- "New Queue Features" on page 119
- "New Uses for LDAP in `sendmail`" on page 119
- "New Built-in Mailer Feature" on page 120
- "New Rule Set Features" on page 121
- "New File Locations" on page 122

New Command-Line Options

The following table describes new command-line options for `sendmail`.

TABLE 21–1 New Command-Line Options for `sendmail`

Option	Description
`-G`	Indicates that the message being submitted from the command line is for relaying, not for initial submission. The message is rejected if the addresses are not fully qualified. No canonicalization is done. As noted in the RELEASE NOTES that are part of the `sendmail` distribution available from ftp://ftp.sendmail.org, improperly formed messages might be rejected in future releases.
`-L` *tag*	Sets the identifier used for syslog messages to the supplied *tag*.

TABLE 21–1 New Command-Line Options for `sendmail` *(Continued)*

Option	Description
`-U`	As noted in the RELEASE NOTES that are part of the `sendmail` distribution available from ftp://ftp.sendmail.org, this option is deprecated. Mail user agents should begin using the `-G` argument to indicate that this is a relay submission (the inverse of the `-U` argument).

New and Revised Configuration File Options and Related Topics

This section contains a table of new and revised configuration file options and information on the following related topics.

- "Deprecated Configuration File Options for `sendmail`" on page 107
- "New `ClientPortOptions` Option" on page 108
- "Changes to `DaemonPortOptions` Option" on page 109
- "Additional Arguments for the `PidFile` and `ProcessTitlePrefix` Options" on page 110
- "Changes to the `PrivacyOptions` Option" on page 110
- "Changes to the `Timeout` Option" on page 111

Note – The `sendmail` options described in the following table are typically declared in the configuration file. However, you can also declare them from the command line. When you use the command line, `sendmail` relinquishes its `root` permissions to avoid a security risk.

When you declare these options, use one of the following syntaxes.

```
O OptionName=argument            # for the configuration file
-OOptionName=argument            # for the command line
define(`m4Name',argument)        # for m4 configuration
```

The following table describes new and revised options for `sendmail`.

TABLE 21–2 New and Revised Options for `sendmail`

Option	Description
`ClientPortOption`	For details, see "New `ClientPortOptions` Option" on page 108.

TABLE 21–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`ControlSocketName`	`m4` name: `confCONTROL_SOCKET_NAME` Argument: *filename*. The recommended socket name is `/var/spool/mqueue/.smcontrol`. For security, this UNIX® domain socket must be in a directory that is accessible only by `root`. When set, this new option creates a control socket for daemon management. This option allows an external program to control and query the status of the running `sendmail` daemon by way of a named socket. The socket is similar to the `ctlinnd` interface to the `INN` news server. If not set, no control socket is available.
`DaemonPortOptions`	For details, see "Changes to `DaemonPortOptions` Option" on page 109.
`DataFileBufferSize`	`m4` name: `confDF_BUFFER_SIZE` Argument: *number* The new option controls the maximum size (in bytes) of a memory-buffered data (`df`) file before a disk-based file is used. The default is `4096` bytes. No changes should be necessary for the Solaris operating environment.
`DeadLetterDrop`	`m4` name: `confDEAD_LETTER_DROP` Argument: *filename* This new option, which you should not need to set, defines the location of the system-wide `dead.letter` file, formerly hard-coded to `/usr/tmp/dead.letter`.
`DontBlameSendmail`	A new argument called, `NonRootSafeAddr`, has been added. When `sendmail` does not have enough privileges to run a `.forward` program or deliver to a file as the owner of that file, addresses are marked unsafe. Furthermore, if `RunAsUser` is set, users cannot use programs or deliver to files in their `.forward` programs. To resolve these problems, use the new argument, `NonRootSafeAddr`.
`DontProbeInterfaces`	`m4` name: `confDONT_PROBE_INTERFACES` Argument: `true` or `false`. The default is `false`. If it is set, `sendmail` does not insert the names and addresses of any local interfaces into class `w` (`$=w`). Therefore, you must also include some support for these addresses (for example, in a mailertable entry). Otherwise, mail to these interface addresses bounces with a configuration error. However, this option, when it is set, speeds up your startup.

TABLE 21–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`LDAPDefaultSpec`	`m4` name: `confLDAP_DEFAULT_SPEC` Argument: Class switch with appropriate definition (for example, `-h`*host*, `-p`*port*, `-d`*bind DN*). The new option allows a default map specification for LDAP maps. The assigned default settings are used for all LDAP maps unless other individual map specifications are made with the `K` command. Set this option before defining any LDAP maps.
`MaxAliasRecursion`	`m4` name: `confMAX_ALIAS_RECURSION` Argument: *number* The option specifies the maximum depth of alias recursion. The defaults are as follows. `50` for a V1/Sun configuration file, which is not recommended for use `10` for any other version of the configuration file
`MaxHeadersLength`	`m4` name: `confMAX_HEADERS_LENGTH` Argument: *number* The option specifies a maximum length for the sum of all headers and can be used to prevent a denial-of-service attack. The default is 32768. Note that a warning is issued if a value less than 16384 is used. You should not need to change the default value for the Solaris operating environment.
`MaxMimeHeaderLength`	`m4` name: `confMAX_MIME_HEADER_LENGTH` Argument: *number* The option sets the maximum length of certain MIME header field values to *x* number of characters. Also, for parameters within headers, you can specify a maximum length of *y*. The combined values look like *x/y*. If */y* is not specified, half of *x* is used. If no values are set, the default is `0`, which means no checks are made. This option is intended to protect mail user agents from buffer-overflow attacks. The suggested values are in the range of 256/128 to 1024/256. A warning is issued if values less than 128/40 are used.
`MaxRecipientsPerMessage`	Argument: *number* If it is set, this option allows no more than the specified number of recipients in an SMTP envelope. The minimum argument is `100`. This option can still be declared from both the command line and the configuration file. However, normal users can now set it from the command line to allow the override of messages submitted through `sendmail -bs`. In this instance, `sendmail` does not relinquish its `root` privileges.

TABLE 21–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`PidFile`	`m4` name: `confPID_file` Argument: See "Additional Arguments for the `PidFile` and `ProcessTitlePrefix` Options" on page 110. The new option defines the location of the `pid` file. The file name is macro-expanded before it is opened. The default is `/var/run/sendmail.pid`.
`PrivacyOptions`	For details, see "Changes to the `PrivacyOptions` Option" on page 110.
`ProcessTitlePrefix`	`m4` name: `confPROCESS_TITLE_PREFIX` Argument: See "Additional Arguments for the `PidFile` and `ProcessTitlePrefix` Options" on page 110. The new option specifies a prefix string for the process title that is shown in `/usr/ucb/ps auxww` listings. The string is macro-processed. No changes should be necessary for the Solaris operating environment.
`QueueLA`	`m4` name: `confQUEUE_LA` Argument: *number* The default value has changed from eight to eight times the number of processors online when the system starts. For single-processor machines, this change has no effect. Changing this value overrides the default and prevents the number of processors from being considered. Therefore, the effect of any value changes should be well understood.
`QueueSortOrder`	`m4` name: `confQUEUE_SORT_ORDER` The `host` argument now reverses the host name before sorting, which means domains are grouped to run through the queue together. This improvement provides better opportunities for use of the connection cache, if available. The new `filename` argument sorts the queue by file name, which avoids the opening and reading of each queue file when preparing to run the queue.
`RefuseLA`	`m4` name: `confREFUSE_LA` Argument: *number* The default value has changed from 12 to 12 times the number of processors online when the system starts. For single-processor machines, this change has no effect. A change of this value overrides the default and prevents the number of processors from being considered. Therefore, the effect of any value changes should be well understood.

TABLE 21–2 New and Revised Options for `sendmail` *(Continued)*

Option	Description
`RrtImpliesDsn`	`m4` name: `confRRT_IMPLIES_DSN` Argument: `true` or `false` If the new option is set, a "Return-Receipt-To:" header causes the request of a delivery status notification (DSN), which is sent to the envelope sender, not to the address given in the header.
`SendMimeErrors`	`m4` name: `confMIME_FORMAT_ERRORS` Argument: `true` or `false` The default is now `true`.
`Timeout`	For details, see "Changes to the `Timeout` Option" on page 111.
`TrustedUser`	`m4` name: `confTRUSTED_USER` Argument: *user name* or *user numeric ID* The new option allows you to specify a user name (instead of `root`) to own important files. If this option is set, generated alias databases and the control socket—if it is configured—are automatically owned by this user. This option requires `HASFCHOWN`. For information about `HASFCHOWN`, see "New Compile Flags for `sendmail`" on page 117. Only `TrustedUser`, `root`, and class `t` (`$=t`) users can rebuild the alias map.
`XscriptFileBufferSize`	`m4` name: `confXF_BUFFER_SIZE` Argument: *number* The new option controls the maximum size (in bytes) of a memory-buffered transcript (`xf`) file before a disk-based file is used. The default is `4096` bytes. No changes should be necessary for the Solaris operating environment.

Deprecated Configuration File Options for `sendmail`

The following table describes deprecated configuration file options for `sendmail`.

TABLE 21–3 Deprecated Configuration File Options for `sendmail`

Option	Description
`AutoRebuildAliases`	Because a denial-of-service attack could occur if this option is set, it has been deprecated. Refer to the RELEASE NOTES that are part of the `sendmail` distribution available from ftp://ftp.sendmail.org. A user could `kill` the `sendmail` process while the aliases file is being rebuilt and leave the file in an inconsistent state.
`MeToo`	This option, which now defaults to `True`, has been deprecated. Refer to the RELEASE NOTES that are part of the `sendmail` distribution available from ftp://ftp.sendmail.org.

New `ClientPortOptions` Option

The new `ClientPortOptions` option is for outgoing connections and is similar to the `DaemonPortOptions` option. This option sets the client SMTP options, which are a sequence of *key=value* pairs. To declare this option, use one of the following syntaxes. (For formatting purposes the example includes two pairs. However, you can apply one or more pairs.

```
O ClientPortOptions=pair,pair              # for the configuration file
-OClientPortOptions=pair,pair              # for the command line
define('confCLIENT_OPTIONS','pair,pair')   # note the revised name
                                           # for m4 configuration
```

The following table describes the new keys for this option.

TABLE 21–4 New Keys for `ClientPortOptions`

Key	Description
`Addr`	Specifies the address mask. The value can be a numeric address in dot notation or a network name. If the pair is omitted, the default is `INADDR_ANY`, which allows connections from any network.
`Family`	Specifies the address family. The key's default is `inet` for AF_INET. Other values are `inet6` for AF_INET6, `iso` for AF_ISO, `ns` for AF_NS, and `x.25` for AF_CCITT.
`Listen`	Specifies the size of the listen queue. The key defaults to 10. No changes should be necessary for the Solaris operating environment.
`Port`	Specifies the name and number of the listening port. The key defaults to `smtp`.

TABLE 21–4 New Keys for `ClientPortOptions` *(Continued)*

Key	Description
`RcvBufSize`	Specifies the size of the TCP/IP send buffer. The key has no default value, which means that no size specifications are automatically made. If the option is set to a value greater than zero, then that value is used. You should not need to limit the size of this buffer for the Solaris operating environment.
`Modifier`	Specifies flags for `sendmail`. The flag, `h`, uses the name that corresponds to the outgoing interface address for the `HELO` or `EHLO` commands, whether it was chosen by the connection parameter or by the default.

Changes to `DaemonPortOptions` Option

The following tables describe two new keys for the option and some specific values for one of the new keys, `Modifier`. To declare this option, use one of the following syntaxes. In the example, *pair* refers to *key=value*. For formatting purposes, the example includes two pairs. However, you can apply one or more pairs.

```
O DaemonPortOptions=pair,pair                # for the configuration file
-ODaemonPortOptions=pair,pair                # for the command line
define('confDAEMON_OPTIONS','pair,pair')     # note the revised name
                                            # for m4 configuration
```

Note – To avoid security risks, `sendmail` relinquishes its root permissions when you set this option from the command line.

The following table describes two new keys for the `DaemonPortOptions` option.

TABLE 21–5 New Keys for `DaemonPortOptions`

Key	Description
`Name`	Specifies a user-definable name for `sendmail` and is used for error messages and for logging. The default is `MTA`.
`Modifier`	Specifies values for `sendmail` that can be listed in a sequence without delimiters. For a list of values, see Table 21–6.

The following table describes the values for the new `Modifier` key.

TABLE 21–6 `Modifier` Key Values for `DaemonPortOptions`

Value	Description
`C`	Does not perform host name canonification.
`E`	Disallows the `ETRN` command.
`a`	Requires authentication.
`b`	Binds to the interface that receives the mail.
`c`	Performs host name canonification. Use this value only in configuration file declarations.
`f`	Requires fully qualified host names. Use this value only in configuration file declarations.
`h`	Uses the interface's name for the outgoing `HELO` command.
`u`	Allows unqualified addresses. Use this value only in configuration file declarations.

Additional Arguments for the `PidFile` and `ProcessTitlePrefix` Options

The following table describes additional macro-processed arguments for the `PidFile` and `ProcessTitlePrefix` options. For more information about these options, see Table 21–2.

TABLE 21–7 Arguments for the `PidFile` and `ProcessTitlePrefix` Options

Macro	Description
`${daemon_addr}`	Provides daemon address (for example, 0.0.0.0)
`${daemon_family}`	Provides daemon family (for example, inet, inet6, and so forth)
`${daemon_info}`	Provides daemon information (for example, SMTP+queueing@00:30:00)
`${daemon_name}`	Provides daemon name (for example, MSA)
`${daemon_port}`	Provides daemon port (for example, 25)
`${queue_interval}`	Provides queue run interval (for example, 00:30:00)

Changes to the `PrivacyOptions` Option

New and revised arguments for `PrivacyOptions` (`popt`) are described in the following table. You can declare this option from the command line without `sendmail` relinquishing its `root` privilege. To declare this `sendmail` option, use one of the following syntaxes.

```
O PrivacyOptions=argument                # for the configuration file
-OPrivacyOptions=argument                # for the command line
define(‘confPRIVACY_FLAGS’,‘argument’)   # note the revised name
                                          # for m4 configuration
```

The following table provides descriptions of new and revised arguments for the `PrivacyOptions` option.

TABLE 21–8 New and Revised Arguments for `PrivacyOptions`

Argument	Description
`goaway`	The `noetrn` and `noreceipts` flags are no longer accepted.
`nobodyreturn`	The argument instructs `sendmail` not to include the body of the original message in delivery status notifications.
`noreceipts`	When the argument is set, delivery status notification (DSN) is not announced.

Changes to the `Timeout` Option

The following table provides information about the changes to the `Timeout` option. Specifically, this `sendmail` option has some new keywords and a new value for `ident`. In the Solaris operating environment, you should not need to change the default values for the keywords that are listed in the table. However, if you choose to make a change, use the *keyword=value* syntax. The *value* is a time interval. Refer to the following examples.

```
O Timeout.keyword=value    # for the configuration file
-OTimeout.keyword=value    # for the command line
define(‘m4_name’, value) # for m4 configuration
```

Note – To avoid security risks, `sendmail` relinquishes its root permissions when you set this option from the command line.

TABLE 21–9 New and Revised Settings for `Timeout`

Keyword	Default Value	Description
`control`	`2m`	`m4` name: `confTO_CONTROL` Limits the total time that is dedicated to satisfying a control socket request.

TABLE 21–9 New and Revised Settings for `Timeout` *(Continued)*

Keyword	Default Value	Description
`ident`	`5s`	`m4` name: `confTO_IDENT` Defaults to 5 seconds—instead of 30 seconds—to prevent the common delays that are associated with mailing to a site that drops IDENT packets.
`queuereturn`	`5d`	`m4` name: `confTO_QUEUERETURN` Includes the value `now`, which immediately bounces entries from the queue without a delivery attempt.
`resolver.retrans`	*varies*	`m4` name: `confTO_RESOLVER_RETRANS` Sets the resolver's retransmission time interval (in seconds), which applies to `resolver.retrans.first` and `resolver.retrans.normal`.
`resolver.retrans.first`	*varies*	`m4` name: `confTO_RESOLVER_RETRANS_FIRST` Sets the resolver's retransmission time interval (in seconds) for the first attempt to deliver a message.
`resolver.retrans.normal`	*varies*	`m4` name: `confTO_RESOLVER_RETRANS_NORMAL` Sets the resolver's retransmission time interval (in seconds) for all resolver lookups, except the first delivery attempt.
`resolver.retry`	*varies*	`m4` name: `confTO_RESOLVER_RETRY` Sets the number of times to retransmit a resolver query, which applies to `Timeout.resolver.retry.first` and `Timeout.resolver.retry.normal`.
`resolver.retry.first`	*varies*	`m4` name: `confTO_RESOLVER_RETRY_FIRST` Sets the number of times to retransmit a resolver query for the first attempt to deliver a message.
`resolver.retry.normal`	*varies*	`m4` name: `confTO_RESOLVER_RETRY_NORMAL` Sets the number of times to retransmit a resolver query for all resolver lookups, except the first delivery attempt.

New Defined Macros for `sendmail`

The following table describes new macros that are reserved for use by the `sendmail` program. Their values are assigned internally.

TABLE 21–10 Defined Macros for `sendmail`

Macro	Description
`${auth_authen}`, `${auth_type}`, `${auth_author}`	Holds the client's authentication credentials, the mechanism used for authentication, and the authorization identity—the `AUTH=` parameter, if supplied.
`${client_resolve}`	Holds the result of the resolve call for `${client_name}`: `OK`, `FAIL`, `FORGED`, or `TEMP`.
`${deliveryMode}`	Specifies the current delivery mode `sendmail` is using, instead of the value of the `DeliveryMode` option.
`${dsn_notify}`, `${dsn_envid}`, `${dsn_ret}`	Holds the corresponding DSN parameter values.
`${if_addr}`	Provides the interface's address for the incoming connection if the interface does not belong to the loopback net. Is especially useful for virtual hosting.
`${if_name}`	Provides the interface's host name for the incoming connection and is especially useful for virtual hosting.
`${load_avg}`	Checks and reports the current average number of jobs in the run queue.
`${msg_size}`	Holds the value of the message size (`SIZE=parameter`) in an ESMTP dialogue before the message has been collected. Thereafter, the macro holds the message size as computed by `sendmail` and is used in `check_compat`.
`${ntries}`	Holds the number of delivery attempts.
`${rcpt_mailer}`, `${rcpt_host}`, `${rcpt_addr}`, `${mail_mailer}`, `${mail_host}`, `${mail_addr}`	Holds the results of parsing the `RCPT` and `MAIL` arguments—that is, the resolved RHS triplet from the mail delivery agent (`$#`*mailer*), the host (`$@`*host*), and the user (`$:`*addr*).

New Macros Used to Build the `sendmail` Configuration File

The following table describes new macros that are used to build the `sendmail` configuration file.

TABLE 21–11 New Macros Used to Build the `sendmail` Configuration File

Flag	Description
`LOCAL_MAILER_EOL`	Overrides the default end-of-line string for the local mailer.
`LOCAL_MAILER_FLAGS`	Adds `Return-Path:` header by default.
`MAIL_SETTINGS_DIR`	Contains the path (including the trailing slash) for the mail settings directory.
`MODIFY_MAILER_FLAGS`	Improves the `*_MAILER_FLAGS`. This macro sets, adds, or deletes flags.
`RELAY_MAILER_FLAGS`	Defines additional flags for the relay mailer.
`USENET_MAILER_FLAGS`	Is not a local mailer. Therefore, the `l` flag has been removed.

New and Revised `m4` Configuration Macros for `sendmail` and Related Topics

This section contains a table of new and revised `m4` configuration macros for `sendmail` and descriptions of the following.

- "New and Revised `FEATURE()` Declarations" on page 115
- "Revised `MAILER()` Declaration for `m4` Configuration" on page 117

Generally, the syntax for declaring the macros that are described in the following table is as shown.

symbolic_name`('`*value*`')`

TABLE 21–12 New and Revised `m4` Configuration Macros for `sendmail`

`m4` Macro	Description
`FEATURE()` declarations	For details, refer to "New and Revised `FEATURE()` Declarations" on page 115.
`LOCAL_DOMAIN()`	This macro adds entries to class `w` (`$=w`).
`MASQUERADE_EXCEPTION()`	A new macro that defines hosts or subdomains that cannot be masqueraded.
`SMART_HOST()`	You can now use this macro for bracketed addresses, such as `user@[`*host*`]`.

TABLE 21–12 New and Revised `m4` Configuration Macros for `sendmail` *(Continued)*

`m4` Macro	Description
`TRUST_AUTH_MECH()`	If SMTP `AUTH` is used, then relaying is allowed for any user who is authenticated as a "trusted" mechanism. This means the mechanism has been defined in the TRUST_AUTH_MECH(*'list_of_mechanisms'*) declaration.
`VIRTUSER_DOMAIN()` or `VIRTUSER_DOMAIN_FILE()`	When these macros are used, include `$={VirtHost}` in `$=R`. As a reminder, `$=R` is the set of host names that are allowed to relay.

New and Revised `FEATURE()` Declarations

The following table describes new and revised keywords for `m4 FEATURE()` declarations. To declare a feature in a `.mc` file, use the syntax from the following example.

`FEATURE(`*`'key_word'`*`, `*`'argument'`*`)`

The following table describes which keywords need *arguments*.

TABLE 21–13 New and Revised Keywords for `FEATURE()` Declarations

Keyword	Description
`delay_checks`	Argument: `friend`, which enables a spam-friend test, or `hater`, which enables spam-hater test. A new keyword that delays all checks. By using `FEATURE('delay_checks')`, the rule sets `check_mail` and `check_relay` are not called when a client connects or issues a `MAIL` command, respectively. Instead, these rule sets are called by the `check_rcpt` rule set. For details, refer to the `/usr/lib/mail/README` file.
`dnsbl`	A new keyword that accepts up to two arguments: DNS server name Rejection message You can include this keyword multiple times.
`generics_entire_domain`	Argument: None A new keyword that you can also use to apply `genericstable` to subdomains of `$=G`.
`ldap_routing`	Argument: For details, refer to the LDAP ROUTING section in `/usr/lib/mail/README`. A new keyword that implements LDAP address routing.

TABLE 21–13 New and Revised Keywords for `FEATURE()` Declarations *(Continued)*

Keyword	Description
`local_lmtp`	Argument: Path name of an LMTP-capable mailer. The default is `mail.local`, which is LMTP-capable in this Solaris release. A keyword that now sets the delivery status notification (DSN) diagnostic-code type for the local mailer to the proper value of `SMTP`.
`nocanonify`	Argument: `canonify_hosts` or nothing A keyword that now includes the following features. Permits a list of domains, as specified by `CANONIFY_DOMAIN` or `CANONIFY_DOMAIN_FILE`, to be passed to the `$[` and `$]` operators for canonification. Permits addresses that have only a host name, such as `<user@host>`, to be canonified, if `canonify_hosts` is specified as its parameter. Adds a trailing dot to addresses with more than one component.
`no_default_msa`	Argument: None A new keyword that turns off `sendmail`'s default setting from `m4`–generated configuration files to listen on several different ports, an implementation of RFC 2476.
`nouucp`	Argument: `reject`, which does not allow the `!` token, or `nospecial`, which does allow the `!` token. A keyword that determines whether or not to allow the `!` token in the local part of an address.
`nullclient`	Argument: None A keyword that now provides the full rule sets of a normal configuration, allowing anti-spam checks to be performed.
`relay_mail_from`	Argument: The *domain* is an optional argument. A new keyword that allows relaying if the mail sender is listed as a `RELAY` in the access map and is tagged with the `From:` header line. If the optional *domain* argument is given, the domain portion of the mail sender is also checked.

TABLE 21–13 New and Revised Keywords for `FEATURE()` Declarations *(Continued)*

Keyword	Description
`virtuser_entire_domain`	Argument: None A keyword that you can now use to apply `$={VirtHost}`, a new class for matching `virtusertable` entries that can be populated by `VIRTUSER_DOMAIN` or `VIRTUSER_DOMAIN_FILE`. `FEATURE('virtuser_entire_domain')` can also apply the class `$={VirtHost}` to entire subdomains.

Revised `MAILER()` Declaration for `m4` Configuration

The `MAILER()` declaration specifies support for delivery agents. To declare a delivery agent, use the following syntax.

`MAILER('`*symbolic_name*`')`

In this new version of `sendmail`, the `MAILER('smtp')` declaration now includes an additional mailer, `dsmtp`, which provides on-demand delivery by using the `F=%` mailer flag. The `dsmtp` mailer definition uses the new `DSMTP_MAILER_ARGS`, which defaults to `IPC $h`.

New Compile Flags for `sendmail`

The following table describes new flags that are used to compile `sendmail`. If your configuration requires other flags, you need to download the source and recompile the binary yourself. You can find information about this process at http://www.sendmail.org.

TABLE 21–14 New Flags Used to Compile `sendmail`

Flag	Description
`HASFCHOWN`	Supports the use of `fchown(2)`.
`HASRANDOM`	Supports the use of `rand(3C)`, instead of `random(3C)`.
`MAXINTERFACES`	Indicates the number of interfaces to read when `sendmail` probes for host names and IP addresses for class `w` (`$=w`). The default value is 512.

New Delivery Agent Flags

The following table describes new delivery agent flags, which by default are not set. These single-character flags are Boolean. You can set or unset a flag by including or excluding it in the `F=` statement of your configuration file, as is shown in the following example.

```
Mlocal,     P=/usr/lib/mail.local, F=lsDFMAw5:/|@qSXfmnz9, S=10/30, R=20/40,
Mprog,      P=/bin/sh, F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/,
Msmtp,      P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990,
Mesmtp,     P=[IPC], F=mDFMuXa, S=11/31, R=21, E=\r\n, L=990,
Msmtp8,     P=[IPC], F=mDFMuX8, S=11/31, R=21, E=\r\n, L=990,
Mrelay,     P=[IPC], F=mDFMuXa8, S=11/31, R=61, E=\r\n, L=2040,
```

TABLE 21–15 New Mailer Flags

Flag	Description
`%`	Mailers that use this flag do not attempt delivery to the initial recipient of a message or to queue runs unless the queued message is selected by using an `ETRN` request or one of the following queue options: `-qI`, `-qR`, or `-qS`.
`6`	This flag allows mailers to strip headers to seven bit.

New Equates for Delivery Agents

The following table describes new equates that you can use with the `M` delivery agent definition command. The following syntax shows you how to append new equates or new arguments to those that already exist in the configuration file.

Magent_name, *equate*, *equate*, ...

The following example includes the new `W=` equate, which specifies the maximum time to wait for the mailer to return after all data has been sent.

```
Msmtp, P=[IPC], F=mDFMuX, S=11/31, R=21, E=\r\n, L=990, W=2m
```

When you modify the definition of a value for `m4` configuration, use the syntax that is provided in the following example.

```
define('SMTP_MAILER_MAXMSGS', '1000')
```

The preceding example limits the number of messages that are delivered per connection on an `smtp` mailer to 1000.

Note – Typically, you modify the equate definitions in the `mailer` directory only when you fine tune.

TABLE 21–16 New Equates for Delivery Agents

Equate	Description
`/=`	Argument: Path to a directory Specifies a directory to `chroot()` into before the mailer program is executed.

TABLE 21–16 New Equates for Delivery Agents *(Continued)*

Equate	Description
m=	Argument: Any of the following m4 values that have previously been defined with the define() routine. SMTP_MAILER_MAXMSGS, for the smtp mailer LOCAL_MAILER_MAXMSGS, for the local mailer RELAY_MAILER_MAXMSGS, for the relay mailer Limits the number of messages that are delivered per connection on an smtp, local, or relay mailer.
W=	Argument: An increment of time Specifies the maximum time to wait for the return of the mailer after all data has been sent.

New Queue Features

The following list provides details about new queue features.

- The update supports multiple queue directories. To use multiple queues, supply a QueueDirectory option value in the configuration file that ends with an asterisk (*), as is shown in the following example.

  ```
  O QueueDirectory=/var/spool/mqueue/q*
  ```

  The option value, /var/spool/mqueue/q*, uses all of the directories (or symbolic links to directories) that begin with "q" as queue directories. Do not change the queue directory structure while sendmail is running. Queue runs create a separate process for running each queue unless the verbose flag (-v) is used on a non-daemon queue run. The new items are randomly assigned to a queue.
- The new queue file-naming system uses file names that are guaranteed to be unique for 60 years. This system allows queue IDs to be assigned without complex file-system locking and makes it easy for queued items to be moved between queues.

New Uses for LDAP in sendmail

The following list describes changes in the use of the Lightweight Directory Access Protocol (LDAP) with sendmail.

- As noted in the RELEASE NOTES that are part of the sendmail distribution available from ftp://ftp.sendmail.org, the LDAPX map has been renamed to LDAP. Use the following syntax for LDAP.

  ```
  Kldap ldap options
  ```

- The update supports the return of multiple values for a single LDAP lookup. Place the values to be returned in a comma-separated string with the `-v` option, as is shown.

  `Kldap ldap -v"`*`mail`*`,`*`more_mail`*`"`

- If no LDAP attributes are specified in an LDAP map declaration, all attributes that are found in the match are returned.
- This version prevents commas in quoted key and value strings in the specifications of the LDAP alias file from breaking up a single entry into multiple entries.
- Instead of using the `%s` token to parse an LDAP filter specification, you can use the new token, `%0`, to encode the key buffer. The `%0` token applies a literal meaning to LDAP special characters.

  The following example shows how these tokens differ for a lookup on "`*`."

TABLE 21–17 Comparison of Tokens

LDAP Map Specification	Specification Equivalent	Result
`-k"uid=%s"`	`-k"uid=*"`	Matches any record with a user attribute
`-k"uid=%0"`	`-k"uid=\2A"`	Matches a user with the name "`*`"

The following table describes new LDAP map flags.

TABLE 21–18 New LDAP Map Flags

Flag	Description
`-1`	Requires a single match to be returned. If more than one match is returned, the results are the equivalent of no records being found.
`-r` *never* | *always* | *search* | *find*	Sets the LDAP alias dereference option.
`-Z` *size*	Limits the number of matches to return.

New Built-in Mailer Feature

The old `[TCP]` built-in mailer is now deprecated. Use the `P=[IPC]` (interprocessor communications) built-in mailer instead. The `[IPC]` built-in mailer now allows delivery to a UNIX domain socket on systems that support it. You can use this mailer with `LMTP` delivery agents that listen on a named socket. An example mailer might look like the following.

```
Mexecmail, P=[IPC], F=lsDFMmnqSXzA5@/:|, E=\r\n,
S=10, R=20/40, T=DNS/RFC822/X-Unix, A=FILE /var/run/lmtpd
```

The first mailer argument in the `[IPC]` mailer is now checked for a legitimate value. The following table provides possible values for the first mailer argument.

TABLE 21–19 Possible Values for the First Mailer Argument

Value	Description
A=FILE	Used for UNIX domain socket delivery
A=TCP	Used for TCP/IP connections
A=IPC	Scheduled for deprecation in a future version

New Rule Set Features

The following table lists the new rule sets and describes what they do.

TABLE 21–20 New Rule Sets

Set	Description
check_eoh	Correlates information that is gathered between headers and checks for missing headers. This rule set is used with the macro storage map and is called after all of the headers have been collected.
check_etrn	Uses the ETRN command (such as check_rcpt uses RCPT).
check_expn	Uses the EXPN command (such as check_rcpt uses RCPT).
check_vrfy	Uses the VRFY command (such as check_rcpt uses RCPT).
trust_auth	Determines whether a given AUTH= parameter of the MAIL command should be trusted.

The following list describes new rule set features.

- Numbered rule sets are also named, but they can still be accessed by their numbers.
- The H header configuration file command allows for a default rule set to be specified for header checks. This rule set is called only if the individual header has not been assigned its own rule set.
- Comments in rule sets (that is, text within parentheses) are not removed if the configuration file version is nine or greater. For example, the following rule matches the input token (1), but does not match the input token.

```
R$+ (1)         $@ 1
```

- sendmail accepts the SMTP RSET command even when it rejects commands because of TCP wrappers or the check_relay rule set.
- You receive a warning if you set the OperatorChars option multiple times. Also, do not set OperatorChars after the rule sets are defined.
- The name of the rule set, as well as its lines, are ignored if an invalid rule set is declared. The rule set lines will not be added to S0.

New File Locations

Please note the new locations for the following files.

- The `helpfile` is now located in `/etc/mail/helpfile`. The old name (`/etc/mail/sendmail.hf`) has a symbolic link that points to the new name.
- The `trusted-users` file is now located in `/etc/mail/trusted-users`. During an upgrade, if the old name (`/etc/mail/sendmail.ct`) is detected, but not the new name, then a hard link from the old name to the new name is created. Otherwise, nothing is done. The default content is `root`.
- The `local-host-names` file is now located in `/etc/mail/local-host-names`. During an upgrade, if the old name (`/etc/mail/sendmail.cw`) is detected, but not the new name, then a hard link from the old name to the new name is created. Otherwise, nothing is done. The default content is zero length.

Changes to `mail.local`

The following table describes the new command-line options for the `mail.local` program, which is used by `sendmail` as a delivery agent for local mail.

TABLE 21–21 New Command-Line Options for `mail.local`

Option	Description
`-7`	Prevents the local mail transfer protocol (LMTP) mode from advertising 8BITMIME support in the `LHLO` response
`-b`	Causes a permanent error instead of a temporary error if a mailbox exceeds its quota

`mail.local` is the default for LMTP mode. However, for this release, if you choose to use `mail.local` as the local delivery agent without being in LMTP mode, you need to do one of the following to set the `S` flag.

Use the following syntax for the configuration file.

```
MODIFY_MAILER_FLAGS('LOCAL', '+S')      # for the configuration file
```

Alternately, perform the following two steps for `m4` configuration.

```
define('MODIFY_MAILER_FLAGS', 'S')dnl   # first step
MAILER(local)dnl                        # second step
```

Note – `MODIFY_MAILER_FLAGS` is a new macro that is used to build the configuration file. For details, refer to "New Macros Used to Build the `sendmail` Configuration File" on page 113.

Changes to `mailstats`

The `mailstats` program, which provides statistics on mailer usage, comes with the `sendmail` program. The following table describes new options in `mailstats`.

TABLE 21–22 New `mailstats` Options

Option	Description
`-C` *filename*	Specifies a `sendmail` configuration file
`-p`	Provides clear statistics in a program-readable mode

Changes to `makemap`

The `makemap` command creates database files for `sendmail`. The following table describes new `makemap` options. When you declare options, use the following syntax.

`makemap` *options class filename*

When you use the preceding syntax, remember the following.

- *options* are preceded by a dash (for example, `-dN`).
- *class* specifies the type of database (for example, `btree`, `dbm`, or `hash`).
- *filename* specifies the full path (or relative name) for the database file.

TABLE 21–23 New `makemap` Options

Option	Description
`-C`	Uses the specified `sendmail` configuration file for finding the `TrustedUser` option
`-c`	Uses the specified `hash` and `btree` cache size

TABLE 21–23 New `makemap` Options *(Continued)*

Option	Description
`-e`	Allows an empty value from the right-hand side (RHS)
`-l`	Lists supported map types
`-u`	Dumps (unmaps) the contents of the database to standard output

Note – If `makemap` is running as root, the ownership of the generated maps is automatically changed to the `TrustedUser` as specified in the `sendmail` configuration file. For more information about the `TrustedUser` option, refer to Table 21–2.

Other Changes and Features of Interest

The following list describes other changes and features of interest.

- As noted in the RELEASE NOTES that are part of the `sendmail` distribution available from ftp://ftp.sendmail.org, the `XUSR` SMTP command is deprecated. Mail user agents should begin using RFC 2476 `Message Submission` for initial user message submission.
- The `Content-Length:` header is no longer provided in messages that are piped to programs with any version of the Sun configuration files. However, this header is still provided for appended messages and ordinary mailbox deliveries that use any version of the Sun configuration files.
- `sendmail` now accepts connections when disk space is low, but in such situations it allows only `ETRN` commands.
- Entries in the alias file can be continued by putting a backslash directly before the new line.
- The timeout for sending a message by way of SMTP has been changed to check for delivery progress every five minutes. This change detects an inability to send information more quickly and reduces the number of processes that are waiting to time out.
- You can now copy the contents of a class to another class by using the syntax of the following example.

  `C{`*Dest*`}  $=`*Source*

  In the preceding example, all items in class `$=`*Source* are copied into class `$=`*Dest*.

- The maps are no longer optional by default. Also, if there is a problem with a map, you receive an error message.
- Canonification is no longer attempted for any host or domain in class `P` (`$=P`).
- The = equate is not included in an option expansion if no value is associated with the option.
- Route addresses are stripped. For example, `<@a,@b,@c:user@d>` is converted to `<user@d>`.

CHAPTER 22

Migration From Berkeley Internet Name Domain (BIND), Version 8.1.2 to BIND Version 8.2.2, Patch Level 5

Berkeley Internet Name Domain (BIND) has migrated from version 8.1.2 to 8.2.2 in the Solaris 8 4/01 release. The following information supplements information on BIND that is in the *Solaris Naming Administration Guide*.

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*.

BIND Upgrade

BIND new functionality in version 8.2.2 includes:

- `In.named` configuration options. See `conf(4)` man page.
- Extensions to the resolver (3RESOLV) interface that are safe to use in multithreaded applications.
- The addition of the `ndc(1M)` command, which is used to start and stop reconfigure `in.named`, and the `dnskeygen(1M)` command, which is used to create TSIG and DNSSEC keys.

CHAPTER 23

IP Network Multipathing

IP network multipathing been enhanced in the Solaris 8 4/01 release. The following information supplements information on IP network multipathing that is in the *IP Network Multipathing Administration Guide*.

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*.

Detaching Network Adapters

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.

In the Solaris 8 4/01 release, dynamic reconfiguration (DR) uses IP Network Multipathing to decommission a specific network device without impacting existing IP users.

For information about dynamic reconfiguration and IP Network Multipathing, see "Detaching Network Adapters" in the *IP Network Multipathing Administration Guide*. Chapter 2 describes relevant procedures.

CHAPTER 24

Mobile IP Administration

Mobile IP administration has been enhanced in the Solaris 8 4/01 release. The following information supplements information on Mobile IP administration that is in the *Mobile IP Administration Guide*.

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*.

Reverse Tunneling and Private Addresses

Mobile Internet Protocol (IP) enables the transfer of information to and from mobile computers, such as laptop 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.

In the Solaris 8 4/01 release, Mobile IP enables you to set up reverse tunnels. By setting up a reverse tunnel from the mobile node's care-of address to the home agent, you ensure a topologically correct source address for the IP data packet. By using reverse tunnels, you can also assign private addresses to mobile nodes.

For an introduction to Mobile IP with reverse tunneling and the use of private addresses, see "Overview of Mobile IP" in the *Mobile IP Administration Guide*. Chapter 2 addresses the Solaris Mobile IP implementation of these new features.

CHAPTER 25

Managing System Resources Topics

This section provides instructions for writing device drivers in the Solaris environment. This section contains this chapter.

Chapter 26	Summarizes updates in the Solaris accounting software, including a new *Extended* accounting file format and new configuration options

CHAPTER 26

Managing Resources With System Accounting

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

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

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. You can charge resource usage by a task 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 27

Managing System Performance Topics

This section provides instructions for managing system performance in the Solaris environment. This section contains these chapters.

Chapter 28	Describes enhancements to the directory name look-up cache (DNLC) that provide improved performance when you access files in large directories with 1000 or more files
Chapter 29	Summarizes updates to the *Solaris Tunable Parameters Reference Manual* for the Solaris 1/01 release

CHAPTER 28

Improving System Performance With DNLC

The directory name look-up cache (DNLC) is enhanced in the Solaris 8 6/00 software release. The following information supplements information on managing system performance in "System Performance (Overview)" in the *System Administration Guide, Volume 2*.

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

The directory name look-up cache (DNLC) is enhanced to provide improved performance when you access files in large directories with 1000 or more files.

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 feature 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`.

CHAPTER 29

Managing System Tuning for Better Performance

The *Solaris Tunable Parameters Reference Manual* has been enhanced in the Solaris 8 1/01 release.

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*.

Changes to the *Solaris Tunable Parameters Reference Manual*

The *Solaris Tunable Parameters Reference Manual* has been updated for the Solaris 8 1/01 release.

Information on the semsys:seminfo_semmnu parameter has been added to the manual. To view this book, see the *Solaris Tunable Parameters Reference Manual*.