System Administration Guide: Advanced Administration

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817–0403–18
October 2009

Contents

Preface

System Administration Guide: Advanced Administration is part of a set that covers a significant part of the Solaris™ system administration information. This guide includes information for both SPARC® and x86 based systems.

This book assumes that you have installed the SunOS™ 5.10 Operating System. It also assumes that you have set up any networking software that you plan to use.

For the Solaris 10 release, new features that are interesting to system administrators are covered in sections called *What's New in ... ?* in the appropriate chapters.

Note – This Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the Solaris OS: Hardware Compatibility Lists (`http://www.sun.com/bigadmin/hcl`). This document cites any implementation differences between the platform types.

In this document these x86 related terms mean the following:

- "x86" refers to the larger family of 64-bit and 32-bit x86 compatible products.
- "x64" relates specifically to 64-bit x86 compatible CPUs.
- "32-bit x86" points out specific 32-bit information about x86 based systems.

For supported systems, see the *Solaris OS: Hardware Compatibility Lists*.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems that are running the Solaris 10 release. To use this book, you should have 1-2 years of UNIX® system administration experience. Attending UNIX system administration training courses might be helpful.

How the System Administration Guides Are Organized

Here is a list of the topics that are covered by the System Administration Guides.

Book Title	Topics
System Administration Guide: Basic Administration	User accounts and groups, server and client support, shutting down and booting a system, managing services, and managing software (packages and patches)
System Administration Guide: Advanced Administration	Terminals and modems, system resources (disk quotas, accounting, and crontabs), system processes, and troubleshooting Solaris software problems
System Administration Guide: Devices and File Systems	Removable media, disks and devices, file systems, and backing up and restoring data
System Administration Guide: IP Services	TCP/IP network administration, IPv4 and IPv6 address administration, DHCP, IPsec, IKE, Solaris IP filter, Mobile IP, IP network multipathing (IPMP), and IPQoS
System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)	DNS, NIS, and LDAP naming and directory services, including transitioning from NIS to LDAP and transitioning from NIS+ to LDAP
System Administration Guide: Naming and Directory Services (NIS+)	NIS+ naming and directory services
System Administration Guide: Network Services	Web cache servers, time-related services, network file systems (NFS and Autofs), mail, SLP, and PPP
System Administration Guide: Solaris Printing	Solaris printing topics and tasks, using services, tools, protocols, and technologies to set up and administer printing services and printers
System Administration Guide: Security Services	Auditing, device management, file security, BART, Kerberos services, PAM, Solaris Cryptographic Framework, privileges, RBAC, SASL, and Solaris Secure Shell
System Administration Guide: Solaris Containers-Resource Management and Solaris Zones	Resource management topics projects and tasks, extended accounting, resource controls, fair share scheduler (FSS), physical memory control using the resource capping daemon (rcapd), and resource pools; virtualization using Solaris Zones software partitioning technology and `lx` branded zones
Solaris ZFS Administration Guide	ZFS storage pool and file system creation and management, snapshots, clones, backups, using access control lists (ACLs) to protect ZFS files, using ZFS on a Solaris system with zones installed, emulated volumes, and troubleshooting and data recovery

Book Title	Topics
Solaris Trusted Extensions Administrator's Procedures	System administration that is specific to a Solaris Trusted Extensions system
Solaris Trusted Extensions Configuration Guide	Starting with the Solaris 10 5/08 release, describes how to plan for, enable, and initially configure Solaris Trusted Extensions

Related Third-Party Web Site References

Note – Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

The Sun web site provides information about the following additional resources:

- Documentation (`http://www.sun.com/documentation/`)
- Support (`http://www.sun.com/support/`)
- Training (`http://www.sun.com/training/`)

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. To share your comments, go to `http://docs.sun.com` and click Feedback.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

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

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

aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

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

TABLE P–2 Shell Prompts

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

General Conventions

Be aware of the following conventions that are used in this book.

- When following steps or using examples, be sure to type double-quotes ("), left single-quotes (‘), and right single-quotes (’) exactly as shown.
- The key referred to as Return is labeled Enter on some keyboards.

- It is assumed that the root path includes the `/sbin`, `/usr/sbin`, `/usr/bin`, and `/etc` directories, so the steps in this book show the commands in these directories without absolute path names. Steps that use commands in other, less common, directories show the absolute path in the example.

CHAPTER 1

Managing Terminals and Modems (Overview)

This chapter provides overview information for managing terminals and modems.

This is a list of the overview information in this chapter:

- "What's New in Managing Terminals and Modems?" on page 21
- "Terminals, Modems, Ports, and Services" on page 23
- "Tools for Managing Terminals and Modems" on page 25
- "Serial Ports Tool" on page 25
- "Service Access Facility" on page 25

For step-by-step instructions on how to set up terminals and modems with the Serial Ports tool, see Chapter 2, "Setting Up Terminals and Modems (Tasks)."

For step-by-step instructions on how to set up terminals and modems with the Service Access Facility (SAF), see Chapter 3, "Managing Serial Ports With the Service Access Facility (Tasks)."

What's New in Managing Terminals and Modems?

This section describes new or changed features for managing terminals and modems in the Solaris release. For a complete listing a new Solaris features and a description of Solaris releases, see *Solaris 10 What's New*.

SPARC: Coherent Console

Solaris 10 8/07: The coherent console subsystem feature implements a part of the kernel console subsystem to facilitate rendering console output. The coherent console uses the Solaris kernel mechanisms to render console output rather than Programmable Read-Only Memory (PROM) interfaces. This reduces the console rendering dependence on the OpenBoot PROM (OBP). The coherent console uses a kernel-resident framebuffer driver to generate console output. The generated console output is more efficient than using OBP rendering. The coherent console also avoids idling CPUs during the SPARC console output and enhances the user experience.

SPARC: Changes to How $TERM Value for Console Is Set

Solaris 10 8/07: The `$TERM` value is now dynamically derived and depends on the terminal emulator that the console is using. On x86 based systems, the `$TERM` value is `sun-color` because the kernel's terminal emulator is always used.

On SPARC based systems the `$TERM` value is as follows:

`sun-color`	This value is used for `$TERM` if the system uses the kernel's terminal emulator.
`sun`	This value is used for `$TERM` if the system uses the PROM's terminal emulator.

This change does not impact how the terminal type is set for the serial port. You can still use the `svccfg` command to modify the `$TERM` value, as shown in the following example:

```
# svccfg
svc:> select system/console-login
svc:/system/console-login> setprop ttymon/terminal_type = "xterm"
svc:/system/console-login> exit
```

`ttymon` Invocations on the System Console Managed by SMF

Solaris 10: `ttymon` invocations on the system console are managed by SMF. The addition of properties to the `svc:/system/console-login:default` service enables you to specify `ttymon` command arguments with the `svccfg` command. Note that these properties are specific to `ttymon`, not generic SMF properties.

Note – You can no longer customize the `ttymon` invocation in the `/etc/inittab` file.

For step-by-step instructions on how to specify `ttymon` command arguments with SMF, see "How to Set the `ttymon` Console Terminal Type" on page 42.

For a complete overview of SMF, see Chapter 17, "Managing Services (Overview)," in *System Administration Guide: Basic Administration*. For information on the step-by-step procedures that are associated with SMF, see Chapter 18, "Managing Services (Tasks)," in *System Administration Guide: Basic Administration*.

Terminals, Modems, Ports, and Services

Terminals and modems provide both local and remote access to system and network resources. Setting up terminals and modem access is an important responsibility of a system administrator. This section explains some of the concepts behind modem and terminal management in the Solaris Operating System.

Terminal Description

Your system's bitmapped graphics display is not the same as an alphanumeric terminal. An alphanumeric terminal connects to a serial port and displays only text. You don't have to perform any special steps to administer the graphics display.

Modem Description

Modems can be set up in three basic configurations:

- Dial-out
- Dial-in
- Bidirectional

A modem connected to your home computer might be set up to provide *dial-out* service. With dial-out service, you can access other computers from your own home. However, nobody outside can gain access to your machine.

Dial-in service is just the opposite. Dial-in service allows people to access a system from remote sites. However, it does not permit calls to the outside world.

Bidirectional access, as the name implies, provides both dial-in and dial-out capabilities.

Ports Description

A *port* is a channel through which a device communicates with the operating system. From a hardware perspective, a port is a “receptacle” into which a terminal or modem cable might be physically connected.

However, a port is not strictly a physical receptacle, but an entity with hardware (pins and connectors) and software (a device driver) components. A single physical receptacle often provides multiple ports, allowing connection of two or more devices.

Common types of ports include serial, parallel, small computer systems interface (SCSI), and Ethernet.

A *serial port*, using a standard communications protocol, transmits a byte of information bit-by-bit over a single line.

Devices that have been designed according to RS-232-C or RS-423 standards, this include most modems, alphanumeric terminals, plotters, and some printers. These devices can be connected interchangeably, using standard cables, into serial ports of computers that have been similarly designed.

When many serial port devices must be connected to a single computer, you might need to add an *adapter board* to the system. The adapter board, with its driver software, provides additional serial ports for connecting more devices than could otherwise be accommodated.

Services Description

Modems and terminals gain access to computing resources by using serial port software. Serial port software must be set up to provide a particular "service" for the device attached to the port. For example, you can set up a serial port to provide bidirectional service for a modem.

Port Monitors

The main mechanism for gaining access to a service is through a *port monitor*. A port monitor is a program that continuously monitors for requests to log in or access printers or files.

When a port monitor detects a request, it sets whatever parameters are required to establish communication between the operating system and the device requesting service. Then, the port monitor transfers control to other processes that provide the services needed.

The following table describes the two types of port monitors included in the Solaris Operating System.

TABLE 1–1 Port Monitor Types

Man Page	Port Monitor	Description
`listen`(1M)	`listen`	Controls access to network services, such as handling remote print requests prior to the Solaris 2.6 release. The default Solaris Operating System no longer uses this port monitor type.
`ttymon`(1M)	`ttymon`	Provides access to the login services needed by modems and alphanumeric terminals. The Serial Ports tool automatically sets up a `ttymon` port monitor to process login requests from these devices.

You might be familiar with an older port monitor called getty. The new ttymon port monitor is more powerful. A single ttymon port monitor can replace multiple occurrences of getty. Otherwise, these two programs serve the same function. For more information, see the getty(1M) man page.

Tools for Managing Terminals and Modems

The following table lists the tools for managing terminals and modems.

TABLE 1–2 Tools For Managing Terminals and Modems

Managing Terminals and Modems Method	Tool	For More Information
The most comprehensive	Service Access Facility (SAF) commands	"Service Access Facility" on page 25
The quickest setup	Solaris Management Console's Serial Ports tool	Chapter 2, "Setting Up Terminals and Modems (Tasks)," and Solaris Management Console online help

Serial Ports Tool

The Serial Ports tool sets up the serial port software to work with terminals and modems by calling the pmadm command with the appropriate information.

The tool also provides the following:

- Templates for common terminal and modem configurations
- Multiple port setup, modification, or deletion
- Quick visual status of each port

Service Access Facility

The SAF is the tool used for administering terminals, modems, and other network devices.

In particular, the SAF enables you to set up the following:

- ttymon and listen port monitors by using the sacadm command
- ttymon port monitor services by using the pmadm and ttyadm commands
- listen port monitor services by using the pmadm and nlsadmin commands
- Troubleshoot tty devices
- Troubleshoot incoming network requests for printing service

- Troubleshoot the Service Access Controller by using the `sacadm` command

The SAF is an open-systems solution that controls access to system and network resources through `tty` devices and local-area networks (LANs). The SAF is not a program, but a hierarchy of background processes and administrative commands.

◆ ◆ ◆ CHAPTER 2

Setting Up Terminals and Modems (Tasks)

This chapter provides step-by-step instructions for setting up terminals and modems using Solaris Management Console's Serial Ports tool.

For overview information about terminals and modems, see Chapter 1, "Managing Terminals and Modems (Overview)." For overview information about managing system resources, see Chapter 4, "Managing System Resources (Overview)."

For information about the procedures associated with setting up terminals and modems using Solaris Management Console's Serial Ports tool, see "Setting Terminals and Modems (Task Map)" on page 27

Setting Terminals and Modems (Task Map)

Task	Description	For Instructions
Set up a terminal.	Set up a terminal by using the Solaris Management Console Serial Ports tool. Configure the terminal by choosing the appropriate option from the Action menu.	"How to Set Up a Terminal" on page 31
Set up a modem.	Set up a modem by using the Solaris Management Console Serial Ports tool. Configure the modem by choosing the appropriate option from the Action menu.	"How to Set Up a Modem" on page 31

Task	Description	For Instructions
Initialize a port.	To initialize a port, use the Solaris Management Console Serial Ports tool. Choose the appropriate option from the Action menu.	"How to Initialize a Port" on page 32

Setting Up Terminals and Modems With Serial Ports Tool (Overview)

You can set up serial ports with the Solaris Management Console's Serial Ports tool.

Select a serial port from the Serial Ports window and then choose a Configure option from the Action menu to configure the following:

- Terminal
- Modem – Dial–In
- Modem – Dial–Out
- Modem – Dial–In/Dial–Out
- Initialize Only – No Connection

The Configure options provide access to the templates for configuring these services. You can view two levels of detail for each serial port: Basic and Advanced. You can access the Advanced level of detail for each serial port after it is configured by selecting the serial port and selecting the Properties option from the Action menu. After a serial port is configured, you can disable or enable the port with the SAF commands. For information on using the SAF commands, see Chapter 3, "Managing Serial Ports With the Service Access Facility (Tasks)."

For information on using the Serial Ports command–line interface, see the `smserialport`(1M) man page.

Setting Up Terminals

The following table describes the menu items (and their default values) when you set up a terminal by using the Serial Ports tool.

TABLE 2–1 Terminal Default Values

Detail	Item	Default Value
Basic	Port	—
	Description	Terminal

TABLE 2–1 Terminal Default Values *(Continued)*

Detail	Item	Default Value
	Service Status	Enabled
	Baud Rate	`9600`
	Terminal Type	`vi925`
	Login Prompt	`ttyn login:`
Advanced	Carrier Detection	Software
	Option: Connect on Carrier	Not available
	Option: Bidirectional	Available
	Option: Initialize Only	Not available
	Timeout (seconds)	Never
	Port Monitor	`zsmon`
	Service Program	`/usr/bin/login`

Setting Up Modems

The following table describes the three modem templates that are available when you set up a modem using the Serial Ports tool.

TABLE 2–2 Modem Templates

Modem Configuration	Description
Dial-In Only	Users can dial in to the modem but cannot dial out.
Dial-Out Only	Users can dial out from the modem but cannot dial in.
Dial-In and Out (Bidirectional)	Users can either dial in or dial out from the modem.

The following table describes the default values of each template.

TABLE 2–3 Modem Template Default Values

Detail	Item	Modem - Dial-In Only	Modem - Dial-Out Only	Modem - Dial In and Out
Basic	Port Name	—	—	—
	Description	Modem – Dial In Only	Modem – Dial Out Only	Modem – Dial In and Out
	Service Status	Enabled	Enabled	Enabled

TABLE 2–3 Modem Template Default Values *(Continued)*

Detail	Item	Modem - Dial-In Only	Modem - Dial-Out Only	Modem - Dial In and Out
	Baud Rate	9600	`9600`	`9600`
	Login Prompt	`ttyn login:`	`ttyn login:`	`ttyn login:`
Advanced	Carrier Detection	Software	Software	Software
	Option: Connect on Carrier	Not available	Not available	Not available
	Option: Bidirectional	Not available	Not available	Available
	Option: Initialize Only	Not available	Available	Not available
	Timeout (seconds)	Never	Never	Never
	Port Monitor	`zsmon`	`zsmon`	`zsmon`
	Service Program	`/usr/bin/login`	`/usr/bin/login`	`/usr/bin/login`

The following table describes the default values for the Initialize Only template.

TABLE 2–4 Initialize Only - No Connection Default Values

Detail	Item	Default Value
Basic	Port Name	—
	Description	Initialize Only - No Connection
	Service Status	Enabled
	Baud Rate	9600
	Login Prompt	`ttyn login:`
Advanced	Carrier Detection	Software
	Option: Connect on Carrier	Not available
	Option: Bidirectional	Available
	Option: Initialize Only	Available
	Timeout (seconds)	Never
	Port Monitor	`zsmon`
	Service Program	`/usr/bin/login`

How to Set Up a Terminal, a Modem, and Initialize a Port (Tasks)

▼ How to Set Up a Terminal

1. **Start the Solaris Management Console, if it's not already running.**

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

   For information on starting the Solaris Management Console, see "Starting the Solaris Management Console" in *System Administration Guide: Basic Administration*.

2. **Click This Computer icon in the Navigation pane.**

3. **Click Devices and Hardware —>Serial Ports.**

   The Serial Ports menu is displayed.

4. **Select the port that will be used with a terminal.**

5. **Choose Configure—>Terminal from the Action menu.**

   The Configure Serial Port window is displayed in Basic Detail mode.

   For a description of the Terminal menu items, see Table 2–1.

6. **Click OK.**

7. **To configure the advanced items, select the port configured as a terminal. Then, select Properties from the Action menu.**

8. **Change the values of template entries, if desired.**

9. **Click OK to configure the port.**

10. **Verify that the terminal service has been added.**

    ```
    $ pmadm -l -s ttyn
    ```

▼ How to Set Up a Modem

1. **Start the Solaris Management Console, if it's not already running.**

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

For information on starting the Solaris Management Console, see "Starting the Solaris Management Console" in *System Administration Guide: Basic Administration*.

2 **Click This Computer icon in the Navigation pane.**

3 **Click Devices and Hardware—>Serial Ports.**

The Serial Ports menu is displayed.

4 **Select the port that will be used with a modem.**

5 **Choose one of the following Configure options from the Action menu.**

a. **Configure—>Modem (Dial In)**

b. **Configure—>Modem (Dial Out)**

c. **Configure—>Modem (Dial In/Out)**

The Configure Serial Port window is displayed in Basic Detail mode.

For a description of the Modem menu items, see Table 2–3.

6 **Click OK.**

7 **To configure the advanced items, select the port configured as a modem. Then, select Properties from the Action menu.**

8 **Change the values of template entries, if desired.**

9 **Click OK to configure the port.**

10 **Verify that the modem service has been configured.**

```
$ pmadm -l -s ttyn
```

▼ How to Initialize a Port

1 **Start the Solaris Management Console, if it's not already running.**

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

For information on starting the Solaris Management Console, see "Starting the Solaris Management Console" in *System Administration Guide: Basic Administration*.

2 **Click This Computer icon in the Navigation pane.**

3 **Click Devices and Hardware—>Serial Ports.**

The Serial Ports menu is displayed.

4 **Select the port to be initialized.**

5 **Choose Configure—>Initialize Only – No Connection**

The Serial Port window is displayed in Basic Detail mode.

For a description of the Initialize Only menu items, see Table 2–4.

6 **Click OK.**

7 **To configure the advanced items, select the port configured as initialize only. Then, select Properties from the Action menu.**

8 **Change the values of template entries, if desired.**

9 **Click OK to configure the port.**

10 **Verify that the modem service has been initialized.**

```
$ pmadm -l -s ttyn
```

Troubleshooting Terminal and Modem Problems

If users are unable to log in over serial port lines after you have added a terminal or modem and set up the proper services, consider the following possible causes of failure:

- Check with the user.

  Malfunctions in terminals and modem use are typically reported by a user who has failed to log in or dial in. For this reason, begin troubleshooting by checking for a problem on the desktop.

  Some common reasons for login failure include:

  - Login ID or password is incorrect
  - Terminal is waiting for X-ON flow control key (Control-Q)
  - Serial cable is loose or unplugged
  - Terminal configuration is incorrect
  - Terminal is shut off or otherwise has no power
- Check the terminal.

  Continue to troubleshoot by checking the configuration of the terminal or modem. Determine the proper *ttylabel* for communicating with the terminal or modem. Verify that the terminal or modem settings match the *ttylabel* settings.

- Check the terminal server.

  If the terminal checks out, continue to search for the source of the problem on the terminal or modem server. Use the `pmadm` command to verify that a port monitor has been configured to service the terminal or modem and that it has the correct *ttylabel* associated with it. For example:

  ```
  $ pmadm -l -t ttymon
  ```

  Examine the `/etc/ttydefs` file and double–check the label definition against the terminal configuration. Use the `sacadm`command to check the port monitor's status. Use `pmadm` to check the service associated with the port the terminal uses.
- Check the serial connection.

  If the Service Access Controller is *starting* the TTY port monitor *and* the following is true:

  - The `pmadm` command reports that the service for the terminal's port is *enabled*.
  - The terminal's configuration matches the port monitor's configuration.

  Then, continue to search for the problem by checking the serial connection. A serial connection comprises serial ports, cables, and terminals. Test each of these parts by using one part with two other parts that are known to be reliable.

  Test all of the following:

  - Serial ports
  - Modems
  - Cables
  - Connectors
- Do not use the Serial Ports tool to modify serial port settings if the serial port is being used as a console. Starting with the Solaris 10 release, invocations of `ttymon` for the console are managed by SMF. For step-by-step instructions on how to change the console terminal type, see "How to Set the `ttymon` Console Terminal Type" on page 42.

For more information on `ttymon` and SMF, see "What's New in Managing Terminals and Modems?" on page 21.

CHAPTER 3

Managing Serial Ports With the Service Access Facility (Tasks)

This chapter describes how to manage serial port services using the Service Access Facility (SAF).

Also included in this chapter is information on how to perform console administration with the Service Management Facility (SMF).

Note – The SAF and SMF are two different tools in the Solaris OS. Starting with the Solaris 10 release, `ttymon` invocations on the system console are now managed by SMF. The SAF tool is still used to administer terminals, modems, and other network devices.

This is a list of the overview information in this chapter.

- "Using the Service Access Facility" on page 36
- "Overall SAF Administration (`sacadm`)" on page 37
- "Port Monitor Service Administration (`pmadm`)" on page 38
- "TTY Monitor and Network Listener Port Monitors" on page 40

For information on the step-by-step procedures that are associated with managing serial ports , see the following:

- "Managing Serial Ports (Task Map)" on page 36
- "Administering `ttymon` services (Task Map)" on page 47

For reference information about the SAF, see "Service Access Facility Administration (Reference)" on page 51.

Managing Serial Ports (Task Map)

Task	Description	For Instructions
Perform console administration.	You might need to perform the following console administration tasks: ■ Set the ttymon console terminal type. Starting with the Solaris 10 release, you must use the svccfg command to specify the ttymon console terminal type. ■ Set the ttymon console terminal baud rate speed.	"How to Set the ttymon Console Terminal Type" on page 42 "How to Set the Baud Rate Speed on the ttymon Console Terminal" on page 43
Add a ttymon port monitor.	Use the sacadm command to add a ttymon port monitor.	"How to Add a ttymon Port Monitor" on page 44
View a ttymon port monitor status.	Use the sacadm command to view ttymon port monitor status.	"How to View ttymon Port Monitor Status" on page 44
Stop a ttymon port monitor.	Use the sacadm command to stop a ttymon port monitor.	"How to Stop a ttymon Port Monitor" on page 45
Start a ttymon port monitor.	Use the sacadm command to start a ttymon port monitor.	"How to Start a ttymon Port Monitor" on page 46
Disable a ttymon port monitor.	Use the sacadm command to disable a ttymon port monitor.	"How to Disable a ttymon Port Monitor" on page 46
Enable a ttymonport monitor.	Use the sacadm command to enable a ttymon port monitor.	"How to Enable a ttymon Port Monitor" on page 46
Remove a ttymon port monitor.	Use the sacadm command to remove a ttymon port monitor.	"How to Remove a ttymon Port Monitor" on page 47

Using the Service Access Facility

You can set up terminals and modems with the Solaris Management Console's Serial Ports tool or the SAF commands.

The SAF is a tool that is used to administer terminals, modems, and other network devices. The top-level SAF program is the Service Access Controller (SAC). The SAC controls port monitors that you administer through the sacadm command. Each port monitor can manage one or more ports.

You administer the services associated with ports through the pmadm command. While services provided through the SAC can differ from network to network, the SAC and its administrative commands, sacadm and pmadm, are network independent.

The following table describes the SAF control hierarchy. The sacadm command is used to administer the SAC, which controls the ttymon and listen port monitors.

The services of ttymon and listen are in turn controlled by the pmadm command. One instance of ttymon can service multiple ports. One instance of listen can provide multiple services on a network interface.

TABLE 3–1 SAF Control Hierarchy

Function	Program	Description
Overall administration	sacadm	Command for adding and removing port monitors
Service Access Controller	sac	SAF's master program
Port monitors	ttymon	Monitors serial port login requests
	listen	Monitors requests for network services
Port monitor service administrator	pmadm	Command for controlling port monitors services
Services	logins, remote procedure calls	Services to which the SAF provides access
Console administration	console login	Console services are managed by the SMF service, svc:/system/console-login:default. This service invokes the ttymon port monitor. Do not use the pmadm or the sacadm command to manage the console. For more information, see "ttymon and the Console Port" on page 40, "How to Set the ttymon Console Terminal Type" on page 42, and "How to Set the Baud Rate Speed on the ttymon Console Terminal" on page 43.

Overall SAF Administration (sacadm)

The sacadm command is the top level of the SAF. The sacadm command primarily is used to add and remove port monitors such as ttymon and listen. Other sacadm functions include listing the current status of port monitors and administering port monitor configuration scripts.

Service Access Controller (SAC Program)

The Service Access Controller program (SAC) oversees all port monitors. A system automatically starts the SAC upon entering multiuser mode.

When the SAC program is invoked, it first looks for, and interprets, each system's configuration script. You can use the configuration script to customize the SAC program environment. This script is empty by default. The modifications made to the SAC environment are inherited by all the “children” of the SAC. This inherited environment might be modified by the children.

After the SAC program has interpreted the per-system configuration script, the SAC program reads its administrative file and starts the specified port monitors. For each port monitor, the SAC program runs a copy of itself, forking a child process. Each child process then interprets its per-port monitor configuration script, if such a script exists.

Any modifications to the environment specified in the per-port monitor configuration script affect the port monitor and will be inherited by all its children. Finally, the child process runs the port monitor program by using the command found in the SAC program administrative file.

SAC Initialization Process

The following steps summarize what happens when SAC is first started:

1. The SAC program is started by the SMF service, `svc:/system/sac:default`.
2. The SAC program reads `/etc/saf/_sysconfig`, the per-system configuration script.
3. The SAC program reads `/etc/saf/_sactab`, the SAC administrative file.
4. The SAC program forks a child process for each port monitor it starts.
5. Each port monitor reads `/etc/saf/pmtag/_config`, the per-port monitor configuration script.

Port Monitor Service Administration (`pmadm`)

The `pmadm` command enables you to administer port monitors' services. In particular, you use the `pmadm` command to add or remove a service and to enable or disable a service. You can also install or replace per-service configuration scripts, or print information about a service.

Each instance of a service must be uniquely identified by a port monitor and a port. When you use the `pmadm` command to administer a service, you specify a particular port monitor with the *pmtag* argument, and a particular port with the *svctag* argument.

For each port monitor type, the SAF requires a specialized command to format port monitor-specific configuration data. This data is used by the `pmadm` command. For `ttymon` and `listen` type port monitors, these specialized commands are `ttyadm` and `nlsadmin`, respectively.

ttymon Port Monitor

Whenever you attempt to log in by using a directly connected modem or alphanumeric terminal, ttymon goes to work. First, the SAC process is started by SMF. Then, the SAC automatically starts the port monitors that are designated in its administrative file, /etc/saf/_sactab. After the ttymon port monitor has been started, it monitors the serial port lines for service requests.

When someone attempts to log in by using an alphanumeric terminal or a modem, the serial port driver passes the activity to the operating system. The ttymon port monitor notes the serial port activity, and attempts to establish a communications link. The ttymon port monitor determines which data transfer rate, line discipline, and handshaking protocol are required to communicate with the device.

After the proper parameters for communication with the modem or terminal are established, the ttymon port monitor passes these parameters to the login program and transfers control to it.

Port Initialization Process

When an instance of the ttymon port monitor is invoked by the SAC, ttymon starts to monitor its ports. For each port, the ttymon port monitor first initializes the line disciplines, if they are specified, and the speed and terminal settings. The values used for initialization are taken from the appropriate entry in the /etc/ttydefs file.

The ttymon port monitor then writes the prompt and waits for user input. If the user indicates that the speed is inappropriate by pressing the Break key, the ttymon port monitor tries the next speed and writes the prompt again.

If *autobaud* is enabled for a port, the ttymon port monitor tries to determine the baud rate on the port automatically. Users must press Return before the ttymon port monitor can recognize the baud rate and print the prompt.

When valid input is received, the ttymon port monitor does the following tasks:

- Interprets the per-service configuration file for the port
- Creates an /etc/utmpx entry, if required
- Establishes the service environment
- Invokes the service associated with the port

After the service terminates, the ttymon port monitor cleans up the /etc/utmpx entry, if this entry exists, and returns the port to its initial state.

Bidirectional Service

If a port is configured for bidirectional service, the ttymon port monitor does the following:

- Allows users to connect to a service
- Allows the uucico, cu, or ct commands to use the port for dialing out, if the port is free
- Waits to read a character before printing a prompt
- Invokes the port's associated service, without sending the prompt message, when a connection is requested, if the connect-on-carrier flag is set

TTY Monitor and Network Listener Port Monitors

Though the SAF provides a generic means for administering any future or third-party port monitors, only two port monitors are implemented in the Solaris Operating System: ttymon and listen.

TTY Port Monitor (ttymon)

The ttymon port monitor is STREAMS-based and does the following:

- Monitors ports
- Sets terminal modes, baud rates, and line disciplines
- Invokes the login process

The ttymon port monitor provides Solaris users the same services that the getty port monitor did under previous versions of SunOS 4.1 software.

The ttymon port monitor runs under the SAC program and is configured with the sacadm command. Each instance of ttymon can monitor multiple ports. These ports are specified in the port monitor's administrative file. The administrative file is configured by using the pmadm and ttyadm commands.

ttymon and the Console Port

Console services are not managed by the Service Access Controller (SAC), nor by any explicit ttymon administration file. ttymon invocations are managed by SMF. As a result, you can no longer invoke ttymon by adding an entry to the /etc/inittab file. A property group with the type, application, and the name ttymon, has been added to the SMF service, svc:/system/console-login:default. The properties within this property group are used by the method script, /lib/svc/method/console-login. This script uses the property values as arguments to the ttymon invocation. Usually, if the values are empty, or if the values are not defined for any of the properties, then the value is not used for ttymon. However, if the ttymon device value is empty, or not set, then /dev/console is used as the default to enable ttymon to run.

The following properties are available under the SMF service, svc:/system/console-login:default:

ttymon/nohangup	Specifies the nohangup property. If set to true, do not force a line hang up by setting the line speed to zero before setting the default or specified speed.
ttymon/prompt	Specifies the prompt string for the console port.
ttymon/terminal_type	Specifies the default terminal type for the console.
ttymon/device	Specifies the console device.
ttymon/label	Specifies the TTY label in the /etc/ttydefs line.

ttymon-Specific Administrative Command (ttyadm)

The ttymon administrative file is updated by the sacadm and pmadm commands, as well as by the ttyadm command. The ttyadm command formats ttymon-specific information and writes it to standard output, providing a means for presenting formatted ttymon-specific data to the sacadm and pmadm commands.

Thus, the ttyadm command does not administer ttymon directly. The ttyadm command complements the generic administrative commands, sacadm and pmadm. For more information, see the ttyadm(1M) man page.

Network Listener Service (listen)

The listen port monitor runs under the SAC and does the following:

- Monitors the network for service requests
- Accepts requests when they arrive
- Invokes servers in response to those service requests

The listen port monitor is configured by using the sacadm command. Each instance of listen can provide multiple services. These services are specified in the port monitor's administrative file. This administrative file is configured by using the pmadm and nlsadmin commands.

The network listener process can be used with any connection-oriented transport provider that conforms to the Transport Layer Interface (TLI) specification. In the Solaris Operating System, listen port monitors can provide additional network services not provided by the inetd service.

Special listen-Specific Administrative Command (nlsadmin)

The listen port monitor's administrative file is updated by the sacadm and pmadm commands, as well as by the nlsadmin command. The nlsadmin command formats listen-specific information and writes it to standard output, providing a means of presenting formatted listen-specific data to the sacadm and pmadm commands.

Thus, the nlsadmin command does not administer listen directly. The command complements the generic administrative commands, sacadm and pmadm.

Each network, configured separately, can have at least one instance of the network listener process associated with it. The nlsadmin command controls the operational states of listen port monitors.

The nlsadmin command can establish a listen port monitor for a given network, configure the specific attributes of that port monitor, and *start* and *kill* the monitor. The nlsadmin command can also report on the listen port monitors on a machine.

For more information, see the nlsadmin(1M) man page.

Administering ttymon Port Monitors

Console administration for ttymon is now managed by SMF. Use the svccfg command to set ttymon system console properties. Continue to use the SAF command, sacadm, to add, list, remove, kill, start, enable, disable, enable, and remove ttymon port monitors.

▼ How to Set the ttymon Console Terminal Type

This procedure shows how to change the console terminal type by using the svccfg command.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Run the svccfg command to set the property for the service instance that you want to change.**

   ```
   # svccfg -s console-login setprop ttymon/terminal_type = "xterm"
   ```

   where xterm is an example of a terminal type that you might want to use.

3 **(Optional) Restart the service instance.**

```
# svcadm restart svc:/system/console-login:default
```

Caution – If you choose to restart the service instance immediately, you are logged out of the console. If you do not restart the service instance immediately, the property changes apply at the next login prompt on the console.

▼ How to Set the Baud Rate Speed on the ttymon Console Terminal

This procedure shows how to set the baud rate speed on the ttymon console terminal. Support for console speeds on x86 based systems are dependent on the specific platform.

The following are supported console speeds for SPARC based systems:

- 9600 bps
- 19200 bps
- 38400 bps

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Use the eeprom command to set a baud rate speed that is appropriate for your system type.**

```
# eeprom ttya-mode=baud-rate,8,n,1,-
```

For example, to change the baud rate on an x86 based system's console to 38400, type:

```
# eeprom ttya-mode=38400,8,n,1,-
```

3 **Change the console line in the /etc/ttydefs file as follows.**

```
console baud-rate hupcl opost onlcr:baud-rate::console
```

4 **Make the following additional changes for your system type.**

Note that these changes are platform-dependent.

- **On SPARC based systems:** Change the baud rate speed in the /kernel/drv/options.conf file.

  Use the following command to change the baud rate to 9600.

```
# 9600             :bd:
ttymodes="2502:1805:bd:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

Use the following command to change the baud rate speed to 19200.

```
# 19200           :be:
ttymodes="2502:1805:be:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

Use the following command to change the baud rate speed to 38400.

```
# 38400           :bf:
ttymodes="2502:1805:bf:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

- **On x86 based systems:** Change the console speed if the BIOS serial redirection is enabled. The method that you use to change the console speed is platform-dependent.

▼ How to Add a `ttymon` Port Monitor

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Add a `ttymon` port monitor.**

   ```
   # sacadm -a -p mbmon -t ttymon -c /usr/lib/saf/ttymon -v ‘ttyadm
   -V‘ -y "TTY Ports a & b"
   ```

   `-a` Specifies the *add* port monitor option.

   `-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

   `-t` Specifies the port monitor *type* as `ttymon`.

   `-c` Defines the *command* string used to start the port monitor.

   `-v` Specifies the *version* number of the port monitor.

   `-y` Defines a comment to describe this instance of the port monitor.

▼ How to View `ttymon` Port Monitor Status

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **View the status of a `ttymon` port monitor.**

   ```
   # sacadm -l -p mbmon
   ```

`-l` Specifies the *list* port monitor status flag.

`-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

Example 3–1 Viewing `ttymon` Port Monitor Status

This example shows how to view a port monitor named, `mbmon`.

```
# sacadm -l -p mbmon
PMTAG   PMTYPE   FLGS RCNT STATUS    COMMAND
mbmon   ttymon   -    0    STARTING  /usr/lib/saf/ttymon #TTY Ports a & b
```

`PMTAG` Identifies the port monitor name, `mbmon`.

`PMTYPE` Identifies the port monitor type, `ttymon`.

`FLGS` Indicates whether the following flags are set:

- d — Do not enable the new port monitor.
- x — Do not start the new port monitor.
- dash (-) — No flags are set.

`RCNT` Indicates the return count value. A return count of `0` indicates that the port monitor is not to be restarted if it fails.

`STATUS` Indicates the current status of the port monitor.

`COMMAND` Identifies the command used to start the port monitor.

`#TTY Ports a & b` Identifies any comment used to describe the port monitor.

How to Stop a `ttymon` Port Monitor

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Stop a `ttymon` port monitor.**

   ```
   # sacadm -k -p mbmon
   ```

   `-k` Specifies the *kill* port monitor status flag.

   `-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

▼ How to Start a ttymon Port Monitor

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Start a killed ttymon port monitor.**

   ```
   # sacadm -s -p mbmon
   ```

   -s   Specifies the *start* port monitor status flag.

   -p   Specifies the *pmtag* mbmon as the port monitor tag.

▼ How to Disable a ttymon Port Monitor

Disabling a port monitor prevents new services from starting, without affecting existing services.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Disable a ttymon port monitor.**

   ```
   # sacadm -d -p mbmon
   ```

   -d   Specifies the *disable* port monitor status flag.

   -p   Specifies the *pmtag* mbmon as the port monitor tag.

▼ How to Enable a ttymon Port Monitor

Enabling a ttymon port monitor allows it to service new requests.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Enable a ttymon port monitor.**

   ```
   # sacadm -e -p mbmon
   ```

   -e   Specifies the *enable* port monitor status flag.

-p Specifies the *pmtag* mbmon as the port monitor tag.

How to Remove a ttymon Port Monitor

Removing a port monitor deletes all the configuration files associated with it.

Note – Port monitor configuration files cannot be updated or changed by using the sacadm command. To reconfigure a port monitor, *remove* it and then *add* a new one.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Remove a ttymon port monitor.**

   ```
   # sacadm -r -p mbmon
   ```

   -r Specifies the *remove* port monitor status flag.

   -p Specifies the *pmtag* mbmon as the port monitor tag.

Administering ttymon services (Task Map)

Task	Description	For Instructions
Add a ttymon service.	Use the pmadm command to add a service.	"How to Add a Service" on page 48
View the Status of a TTY Port Service.	Use the pmadmcommand to view the status of a TTY port.	"How to View the Status of a TTY Port Service" on page 49
Enable a port monitor service.	Use the pmadm command with the -e option to enable a port monitor.	"How to Enable a Port Monitor Service" on page 51
Disable a port monitor service.	Use the pmadm command with the -d option to disable a port monitor.	"How to Disable a Port Monitor Service" on page 51

Administering ttymon Services

Use the pmadm command to add services, list the services of one or more ports associated with a port monitor, and enable or disable a service.

▼ How to Add a Service

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Add a standard terminal service to the mbmon port monitor.**

```
# pmadm -a -p mbmon -s a -i root -v ‘ttyadm -V‘ -m "‘ttyadm -i ’Terminal
 disabled’ -l contty -m ldterm,ttcompat -S y -d /dev/term/a
-s /usr/bin/login‘"
```

Note – In this example, the input wraps automatically to the next line. Do not use a Return key or line feed.

-a Specifies the *add* port monitor status flag.

-p Specifies the *pmtag* mbmon as the port monitor tag.

-s Specifies the *svctag* a as the port monitor *service* tag.

-i Specifies the *identity* to be assigned to *svctag* when the service runs.

-v Specifies the *version* number of the port monitor.

-m Specifies the ttymon-specific configuration data formatted by ttyadm.

The preceding pmadm command contains an embedded ttyadm command. The options in this embedded command are as follows:

-b Specifies the *bidirectional* port flag.

-i Specifies the *inactive* (disabled) response message.

-l Specifies which TTY *label* in the /etc/ttydefs file to use.

-m Specifies the STREAMS *modules* to push before invoking this service.

-d Specifies the full path name to the *device* to use for the TTY port.

-s Specifies the full path name of the *service* to invoke when a connection request is received. If arguments are required, enclose the command and its arguments in quotation marks (").

▼ How to View the Status of a TTY Port Service

Use the pmadm command as shown in this procedure to list the status of a TTY port or all the ports that are associated with a port monitor.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **List one service of a port monitor.**

   ```
   # pmadm -l -p mbmon -s a
   ```

   -l Lists service information on the system.

   -p Specifies the *pmtag* mbmon as the port monitor tag.

   -s Specifies the *svctag* a as the port monitor *service* tag.

Example 3–2 Viewing the Status of a TTY Port Monitor Service

This example lists all services of a port monitor.

```
# pmadm -l -p mbmon
PMTAG  PMTYPE  SVCTAG FLAGS ID    <PMSPECIFIC>
mbmon  ttymon  a      -     root  /dev/term/a - - /usr/bin/login - contty
ldterm,ttcompat login:  Terminal disabled tvi925 y  #
```

PMTAG Identifies the port monitor name, mbmon, that is set by using the pmadm -p command.

PMTYPE Identifies the port monitor type, ttymon.

SVCTAG Indicates the service tag value that is set by using the pmadm -s command.

FLAGS Identifies whether the following flags are set by using the pmadm -f command.

- x — Do not enable the service.
- u — Create a utmpx entry for the service.
- dash (-) — No flags are set.

ID Indicates the identity assigned to the service when it is started. This value is set by using the pmadm -i command.

<PMSPECIFIC> *Information*

/dev/term/a Indicates the TTY port path name that is set by using the ttyadm -d command.

`-`

Indicates whether the following flags are set by using the `ttyadm -c -b -h -I -r` command.

- `c` — Sets the connect on carrier flag for the port.
- `b` — Sets the port as bidirectional, allowing both incoming and outgoing traffic.
- `h` — Suppresses an automatic hangup immediately after an incoming call is received.
- `I` — Initializes the port.
- `r`— Forces `ttymon` to wait until it receives a character from the port before it prints the `login:` message.
- dash (-) — No flags are set.

`-`

Indicates a value that is set by using the `ttyadm -r` *count* option. This option determines when `ttymon` displays a prompt after receiving data from a port. If *count* is 0, `ttymon` waits until it receives any character. If *count* is greater than 0, `ttymon` waits until *count* new lines have been received. No value is set in this example.

`/usr/bin/login`

Identifies the full path name of the service to be invoked when a connection is received. This value is set by using the `ttyadm -s` command.

`-`

Identifies the `ttyadm -t` command's time-out value. This option specifies that `ttymon` should close a port if the open on the port succeeds, and no input data is received in *timeout* seconds. There is no time-out value in this example.

`contty`

Identifies the TTY label in the `/etc/ttydefs` file. This value is set by using the `ttyadm -l` command.

`ldterm,ttcompat`

Identifies the STREAMS modules to be pushed. These modules are set by using the `ttyadmin -m` command.

`login: Terminal disabled`

Identifies an inactive message to be displayed when the port is disabled. This message is set by using the `ttyadm -i` command.

`tvi925`

Identifies the terminal type, if set, by using the `ttyadm -T` command. The terminal type is `tvi925` in this example.

`y`

Identifies the software carrier value that is set by using the `ttyadm -S` command. `n` turns the software carrier off. `y` turns the software carrier on. The software carrier is turned on in this example.

#	Identifies any comment specified with the `pmadm -y` command. There is no comment in this example.

▼ How to Enable a Port Monitor Service

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Enable a disabled port monitor service.**

```
# pmadm -e -p mbmon -s a
```

`-e`	Specifies the *enable* flag.
`-p`	Specifies the *pmtag* `mbmon` as the port monitor tag.
`-s`	Specifies the *svctag* `a` as the port monitor *service* tag.

▼ How to Disable a Port Monitor Service

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Disable a port monitor service.**

```
# pmadm -d -p mbmon -s a
```

`-d`	Specifies the *disable* flag.
`-p`	Specifies the *pmtag* `mbmon` as the port monitor tag.
`-s`	Specifies the *svctag* `a` as the port monitor *service* tag.

Service Access Facility Administration (Reference)

This chapter includes reference information for administration of the Service Access Facility.

Files Associated With the SAF

The SAF uses configuration files that can be modified by using the `sacadm` and `pmadm` commands. You should not need to manually edit the configuration files.

File Name	Description
`/etc/saf/_sysconfig`	Per-system configuration script.
`/etc/saf/_sactab`	The SAC's administrative file that contains configuration data for the port monitors that the SAC controls
`/etc/saf/`*pmtag*	Home directory for port monitor *pmtag*
`/etc/saf/`*pmtag*`/_config`	Per-port monitor configuration script for port monitor *pmtag* if it exists
`/etc/saf/`*pmtag*`/`*_pmtab*	Port monitor *pmtag*'s administrative file that contains port monitor-specific configuration data for the services *pmtag* provides
`/etc/saf/`*pmtag*`/`*svctag*	Per-service configuration script for service *svctag*
`/var/saf/log`	The SAC's log file
`/var/saf/`*pmtag*	Directory for files created by *pmtag*, for example, log files

`/etc/saf/_sactab` File

The information in the `/etc/saf/_sactab` file is as follows:

```
# VERSION=1
zsmon:ttymon::0:/usr/lib/saf/ttymon
#
```

`# VERSION=1` Indicates the Service Access Facility version number.

`zsmon` Is the name of the port monitor.

`ttymon` Is the type of port monitor.

`::` Indicates whether the following two flags are set:

- d — Do not enable the port monitor.
- x — Do not start the port monitor. No flags are set in this example.

0 — Indicates the return code value. A return count of 0 indicates that the port monitor is not be restarted if the port monitor fails.

/usr/lib/saf/ttymon — Indicates the port monitor path name.

/etc/saf/*pmtab/_pmtab* File

The /etc/saf/*pmtab/_pmtab* file, such as /etc/saf/zsmon/_pmtab, is similar to the following:

```
# VERSION=1
ttya:u:root:reserved:reserved:reserved:/dev/term/a:I::/usr/bin/login::9600:
ldterm,ttcompat:ttya login\: ::tvi925:y:#
```

# VERSION=1 — Indicates the Service Access Facility version number.

ttya — Indicates the service tag.

x,u — Identifies whether the following flags are set:

- x — Do not enable the service.
- u — Create a utmpx entry for the service.

root — Indicates the identity assigned to the service tag.

reserved — This field is reserved for future use.

reserved — This field is reserved for future use.

reserved — This field is reserved for future use.

/dev/term/a — Indicates the TTY port path name.

/usr/bin/login — Identifies the full path name of the service to be invoked when a connection is received.

:c,b,h,I,r: — Indicates whether the following flags are set:

c — Sets the connect on carrier flag for the port.

b — Sets the port as bidirectional, allowing both incoming and outgoing traffic.

h — Suppresses an automatic hangup immediately after an incoming call is received.

I — Initializes the port.

r — Forces ttymon to wait until it receives a character from the port before ttymon prints the login: message.

9600 — Identifies the TTY label defined in the /etc/ttydefs file.

`ldterm,ttcompat`	Identifies the STREAMS modules to be pushed.
`ttya login\:`	Identifies the prompt to be displayed.
`:y/n:`	Indicates yes or no response.
message	Identifies any inactive (disabled) response message.
`tvi925`	Identifies the terminal type.
`y`	Indicates whether the software carrier is set (`y`/`n`).

Service States

The `sacadm` command controls the states of services. The following table describes the possible states of services.

State	Description
Enabled	*Default state* – When the port monitor is added, the service operates.
Disabled	*Default state* – When the port monitor is removed, the service stops.

To determine the state of any particular service, use the following:

```
# pmadm -l -p portmon-name -ssvctag
```

Port Monitor States

The `sacadm` command controls the states of the `ttymon` and `listen` port monitors. The following table describes the possible port monitor states.

State	Description
`Started`	*Default state* – When the port monitor is added, it is automatically started.
`Enabled`	*Default state* – When the port monitor is added, it is automatically ready to accept requests for service.
`Stopped`	*Default state* – When the port monitor is removed, it is automatically stopped.
`Disabled`	*Default state* – When the port monitor is removed, it automatically continues existing services and refuses to add new services.
`Starting`	*Intermediate state* – The port monitor is in the process of starting.

State	Description
`Stopping`	*Intermediate state* – The port monitor has been manually terminated, but it has not completed its shutdown procedure. The port monitor is on the way to becoming stopped.
`Notrunning`	*Inactive state* – The port monitor has been killed. All ports previously monitored are inaccessible. An external user cannot tell whether a port is `disabled` or `notrunning`.
`Failed`	*Inactive state* – The port monitor is unable to start and remain running.

To determine the state of any particular port monitor, use the following command:

```
# sacadm -l -p portmon-name
```

Port States

Ports can be enabled or disabled depending on the state of the port monitor that controls the ports.

State	Description
Serial (`ttymon`) port states	
`Enabled`	The `ttymon` port monitor sends a prompt message to the port and provides login service to it.
`Disabled`	Default state of all ports if `ttymon` is killed or disabled. If you specify this state, `ttymon` sends out the `disabled` message when it receives a connection request.

◆ ◆ ◆ CHAPTER 4

Managing System Resources (Overview)

This chapter provides a brief description of the system resource management features that are available in the Solaris Operating System and a road map to help you manage system resources.

Using these features, you can display general system information, monitor disk space, set disk quotas and use accounting programs. You can also schedule the `cron` and `at` commands to automatically run routine commands.

This section does not cover information on Solaris resource management that enables you to allocate, monitor, and control system resources in a flexible way.

For information on the procedures that are associated with managing system resources without Solaris resource management, see "Managing System Resources (Road Map)" on page 59.

For information on managing system resources with Solaris resource management, see Chapter 1, "Introduction to Solaris 10 Resource Manager," in *System Administration Guide: Solaris Containers-Resource Management and Solaris Zones*.

What's New in Managing System Resources?

This section describes new or changed features for managing system resources in this Solaris release. For a complete listing of new Solaris features and a description of Solaris releases, see *Solaris 10 What's New*.

New `prtconf` Option to Display Product Names

Solaris 10 1/06: A new `-b` option has been added to the `prtconf` command for the purpose of displaying a system's product name. This option is similar to the `uname -i` command. However, the `prtconf -b` command is specifically designed to determine the marketing name of a product.

The firmware device tree root properties that are displayed by using the -b option to the prtconf command are as follows:

- name
- compatible
- banner-name
- model

To display additional platform- specific output that might be available, use the prtconf -vb command. For more information, see the prtconf(1M) man page and "How to Display a System's Product Name" on page 67.

New psrinfo Command Option to Identify Chip Multithreading Features

Solaris 10: The psrinfo command has been modified to provide information about physical processors, in addition to information about virtual processors. This enhanced functionality has been added to identify chip multithreading (CMT) features. The new -p option reports the total number of physical processors that are in a system. Using the psrinfo -pv command will list all the physical processors that are in the system, as well as the virtual processors that are associated with each physical processor. The default output of the psrinfo command continues to display the virtual processor information for a system.

For more information, see the psrinfo(1M) man page.

For information about the procedures associated with this feature, see "How to Display a System's Physical Processor Type" on page 68.

New localeadm Command

Solaris 10: The new localeadm command allows you to change the locales on your system without reinstalling the OS or manually adding and removing packages. This command also allows you to query your system to determine which locales are installed. To run the localeadm command, you must have superuser privileges or assume an equivalent role through role-based access control (RBAC).

For more information, see the localeadm(1M) man page.

For more information in this guide, see Chapter 5, "Displaying and Changing System Information (Tasks)."

For a complete listing of new Solaris features and a description of Solaris releases, see *Solaris 10 What's New*.

Managing System Resources (Road Map)

Task	Description	Instructions
Displaying and changing system information	Use various commands to display and change system information, such as general system information, the language environment, the date and time, and the system's host name.	Chapter 5, “Displaying and Changing System Information (Tasks)”
Managing disk use	Identify how disk space is used and take steps to remove old and unused files.	Chapter 6, “Managing Disk Use (Tasks)”
Managing quotas	Use UFS file system quotas to manage how much disk space is used by users.	Chapter 7, “Managing UFS Quotas (Tasks)”
Scheduling system events	Use cron and at jobs to help schedule system routines that can include clean up of old and unused files.	Chapter 8, “Scheduling System Tasks (Tasks)”
Managing system accounting	Use system accounting to identify how users and applications are using system resources.	Chapter 9, “Managing System Accounting (Tasks)”
Managing system resources with Solaris Resource Management	Use resource manager to control how applications use available system resources and to track and charge resource usage.	Chapter 1, “Introduction to Solaris 10 Resource Manager,” in *System Administration Guide: Solaris Containers-Resource Management and Solaris Zones*

CHAPTER 5

Displaying and Changing System Information (Tasks)

This chapter describes the tasks that are required to display and change the most common system information.

For information about the procedures associated with displaying and changing system information, see the following:

- "Displaying System Information (Task Map)" on page 61
- "Changing System Information (Task Map)" on page 71

For overview information about managing system resources, see Chapter 4, "Managing System Resources (Overview)."

Displaying System Information (Task Map)

Task	Description	For Instructions
Determine whether a system has 32 bit or 64–bit capabilities enabled.	Use the `isainfo` command to determine whether a system has 32–bit or 64-bit capabilities enabled. For x86 based systems, you can use the `isalist` command to display this information.	"How to Determine Whether a System Has 32–bit or 64–Bit Solaris Capabilities Enabled" on page 63
Display Solaris Release Information	Display the contents of the `/etc/release` file to identify your Solaris release version.	"How to Display Solaris Release Information" on page 66
Display General System Information.	Use the `showrev` command to display general system information.	"How to Display General System Information" on page 66

Task	Description	For Instructions
Display a system's Host ID number.	Use the `hostid` command to display your system's host id.	"How to Display a System's Host ID Number" on page 67
Display a System's product name	Starting with the Solaris 10 1/06 release, you can use the `prtconf -b` command to display the product name of a system.	"How to Display a System's Product Name" on page 67
Display a System's Installed Memory	Use the `prtconf` command to display information about your system's installed memory.	"How to Display a System's Installed Memory" on page 68
Display a system's date and time.	Use the `date` command to display your system's date and time.	"How to Display the Date and Time" on page 68
Display a system's physical processor type.	Use the `psrinfo -p` command to list the total number of physical processors on a system. Use the `psrinfo -pv` command to list all physical processors on a system and the virtual processors that is associated with each physical processor.	"How to Display a System's Physical Processor Type" on page 68
Display a system's logical processor type.	Use the `psrinfo -v` command to display a system's logical processor type.	"How to Display a System's Logical Processor Type" on page 69
Display locales that are installed on a system.	Use the `localeadm` command to display locales that are installed on your system.	"How to Display Locales Installed on a System" on page 69
Determine if a locale is installed on a system.	Use the `-q` option of the `localeadm` command and a locale to determine if a locale is installed on your system.	"How to Determine if a Locale is Installed on a System" on page 70

Displaying System Information

The following table describes commands that enable you to display general system information.

TABLE 5–1 Commands for Displaying System Information

Command	System Information Displayed	Man Page
`date`	Date and time	`date(1)`

TABLE 5–1 Commands for Displaying System Information *(Continued)*

Command	System Information Displayed	Man Page
`hostid`	Host ID number	`hostid(1)`
`isainfo`	The number of bits supported by *native* applications on the running system, which can be passed as a token to scripts	`isainfo(1)`
`isalist`	Processor type for x86 based systems	`psrinfo(1M)`
`localeadm`	Locales installed on the system	`localeadm(1M)`
`prtconf`	System configuration information, installed memory, and product name	`prtconf(1M)`
`psrinfo`	Processor type	`psrinfo(1M)`
`showrev`	Host name, host ID, release, kernel architecture, application architecture, hardware provider, domain, and kernel version	`showrev(1M)`
`uname`	Operating system name, release, version, node name, hardware name, and processor type	`uname(1)`

▼ How to Determine Whether a System Has 32–bit or 64–Bit Solaris Capabilities Enabled

● **Use the `isainfo` command to determine whether a system has 32–bit or 64-bit capabilities enabled.**

```
# isainfo options
```

The `isainfo` command, run without specifying any options, displays the name or names of the native instruction sets for applications supported by the current OS version.

`-v` Prints detailed information about the other options

`-b` Prints the number of bits in the address space of the native instruction set.

`-n` Prints the name of the native instruction set used by portable applications supported by the current version of the OS.

`-k` Prints the name of the instruction set or sets that are used by the OS kernel components such as device drivers and STREAMS modules.

Note – For x86 based systems, the `isalist` command can also be used to display this information.

For more information, see the`isalist(1)` man page.

Example 5–1 SPARC: Determining Whether a System Has 32–Bit or 64–Bit Solaris Capabilities Enabled

The `isainfo` command output for an UltraSPARC system that is running previous releases of the Solaris OS using a 32-bit kernel is displayed as follows:

```
$ isainfo -v
32-bit sparc applications
```

This output means that this system can support only 32–bit applications.

The current release of the Solaris OS only ships a 64–bit kernel on SPARC based systems. The `isainfo` command output for an UltraSPARC system that is running a 64–bit kernel is displayed as follows:

```
$ isainfo -v
64-bit sparcv9 applications
32-bit sparc applications
```

This output means that this system is capable of supporting both 32–bit and 64–bit applications.

Use the `isainfo -b` command to display the number of bits supported by native applications on the running system.

The output from a SPARC based, x86 based, or UltraSPARC system that is running the 32–bit Solaris Operating System is displayed as follows:

```
$ isainfo -b
32
```

The `isainfo` command output from a 64–bit UltraSPARC system that is running the 64–bit Solaris Operating System is displayed as follows:

```
$ isainfo -b
64
```

The command returns `64` only. Even though a 64–bit UltraSPARC system can run both types of applications, 64–bit applications are the best kind of applications to run on a 64–bit system.

Example 5–2 x86: Determining Whether a System Has 32–Bit or 64–Bit Solaris Capabilities Enabled

The `isainfo` command output for an x86 based system that is running the 64-bit kernel is displayed as follows:

```
$ isainfo
amd64 i386
```

This output means that this system can support 64–bit applications.

Use the isainfo `-v` command to determine if an x86 based system is capable of running a 32–bit kernel.

```
$ isainfo -v
64-bit amd64 applications
        fpu tsc cx8 cmov mmx ammx a3dnow a3dnowx fxsr sse sse2
32-bit i386 applications
        fpu tsc cx8 cmov mmx ammx a3dnow a3dnowx fxsr sse sse2
```

This output means that this system can support both 64–bit and 32–bit applications.

Use the `isainfo -b` command to display the number of bits supported by native applications on the running system.

The output from an x86 based system that is running the 32–bit Solaris Operating System is displayed as follows:

```
$ isainfo -b
32
```

The `isainfo` command output from an x86 based system that is running the 64–bit Solaris Operating System is displayed as follows:

```
$ isainfo -b
64
```

You can also use the `isalist` command to determine whether an x86 based system is running in 32–bit or 64–bit mode.

```
$ isalist
amd64 pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

In the preceding example, `amd64` indicates that the system has 64–bit Solaris capabilities enabled.

▼ How to Display Solaris Release Information

- **Display the contents of the `/etc/release` file to identify your Solaris release version.**

```
$ cat /etc/release
                        Solaris 10 s10_51 SPARC
           Copyright 2004 Sun Microsystems, Inc.  All Rights Reserved.
                        Use is subject to license terms.
                            Assembled 21 January 2004
```

▼ How to Display General System Information

- **To display general system information, use the `showrev` command.**

```
$ showrev options
```

`-a` Prints all system revision information available.

`-c (command)` Prints the revision information about command

`-p` Prints only the revision information about patches.

`-R (root_path)` Defines the full path name of a directory to use as the root_path.

`-s (host name)` Performs this operation on the specified host name

`-w` Prints only the OpenWindows revision information.

You can also use the `uname` command to display system information. The following example shows the `uname` command output. The `-a` option displays the operating system name as well as the system node name, operating system release, operating system version, hardware name, and processor type.

```
$ uname
SunOS
$ uname -a
SunOS starbug 5.10 Generic sun4u sparc SUNW,Ultra-5_10
$
```

Example 5–3 Displaying General System Information

The following example shows the `showrev` command output. The `-a` option displays all available system information.

```
$ showrev -a
Hostname: stonetouch
Hostid: 8099dfb9
Release: 5.10
Kernel architecture: sun4u
```

```
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: boulder.Central.Sun.COM
Kernel version: SunOS 5.10 s10_46

OpenWindows version:
Solaris X11 Version 6.6.2 20 October 2003

No patches are installed
```

▼ How to Display a System's Host ID Number

- **To display the host ID number in hexadecimal format, use the `hostid` command.**

Example 5–4 Displaying a System's Host ID Number

The following example shows sample output from the `hostid` command.

```
$ hostid
80a5d34c
```

▼ How to Display a System's Product Name

Solaris 10 1/06: The `-b` option to the `prtconf` command enables you to display a system's product name. For more information on this feature, see the `prtconf`(1M) man page.

- **To display the product name for your system, use the `prtconf` command with the `-b` option.**

Example 5–5 Displaying a System's Product Name

This example shows sample output from the `prtconf -b` command.

```
# prtconf -b
name:  SUNW,Ultra-5_10
model:  SUNW,375-0066
banner-name:  Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
```

This example shows sample output from the `prtconf -vb` command.

```
# prtconf -vb
name:  SUNW,Ultra-5_10
model:  SUNW,375-0066
banner-name:  Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
idprom: 01800800.20a6c363.00000000.a6c363a9.00000000.00000000.405555aa.aa555500
openprom model:  SUNW,3.15
openprom version: ’OBP 3.15.2 1998/11/10 10:35’
```

▼ How to Display a System's Installed Memory

- **To display the amount of memory that is installed on your system, use the `prtconf` command.**

Example 5–6 Displaying a System's Installed Memory

The following example shows sample output from the `prtconf` command. The `grep Memory` command selects output from the `prtconf` command to display memory information only.

```
# prtconf | grep Memory
Memory size: 128 Megabytes
```

▼ How to Display the Date and Time

- **To display the current date and time according to your system clock, use the `date` command.**

Example 5–7 Displaying the Date and Time

The following example shows sample output from the `date` command.

```
$ date
Wed Jan 21 17:32:59 MST 2004
$
```

▼ How to Display a System's Physical Processor Type

- **Use the `psrinfo -p` command to display the total number of physical processors on a system.**

```
$ psrinfo -p
1
```

Use the `psrinfo -pv` command to display information about each physical processor on a system, and the virtual processor associated with each physical processor.

```
$ psrinfo -pv
The UltraSPARC-IV physical processor has 2 virtual processors (8, 520)
The UltraSPARC-IV physical processor has 2 virtual processors (9, 521)
The UltraSPARC-IV physical processor has 2 virtual processors (10, 522)
The UltraSPARC-IV physical processor has 2 virtual processors (11, 523)
The UltraSPARC-III+ physical processor has 1 virtual processor (16)
The UltraSPARC-III+ physical processor has 1 virtual processor (17)
The UltraSPARC-III+ physical processor has 1 virtual processor (18)
The UltraSPARC-III+ physical processor has 1 virtual processor (19)
```

When you use the `psrinfo -pv` command on an x86 based system, the following output is displayed:

```
$ psrinfo -pv
The i386 physical processor has 2 virtual processors (0, 2)
The i386 physical processor has 2 virtual processors (1, 3)
```

▼ How to Display a System's Logical Processor Type

- **Use the `psrinfo -v` command to display information about a system's processor type.**

  ```
  $ psrinfo -v
  ```

  On an x86 based system, use the `isalist` command to display the virtual processor type.

  ```
  $ isalist
  ```

Example 5–8 SPARC: Displaying a System's Processor Type

This example shows how to display information about a SPARC based system's processor type.

```
$ psrinfo -v
Status of virtual processor 0 as of: 04/16/2004 10:32:13
  on-line since 03/22/2004 19:18:27.
  The sparcv9 processor operates at 650 MHz,
  and has a sparcv9 floating point processor.
```

Example 5–9 x86: Displaying a System's Processor Type

This example shows how to display information about an x86 based system's processor type.

```
$ isalist
pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

▼ How to Display Locales Installed on a System

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Display the locales currently installed on your system using the `localeadm` command. The `-l` option displays the locales that are installed on the system. For example:**

```
# localeadm -l
Checking for installed pkgs. This could take a while.

Checking for Australasia region (aua)
(1of2 pkgs)
|......|
.
.
.
The following regions are installed on concordance on Wed Dec 17 15:13:00 MST 2003


POSIX (C)

Central Europe (ceu)
[ Austria, Czech Republic, Germany, Hungary, Poland, Slovakia,
Switzerland (German), Switzerland (French) ]

Done.
```

▼ How to Determine if a Locale is Installed on a System

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Determine if a locale is installed on your system using the `localeadm` command. The `-q` option and a locale queries the system to see if that locale is installed on the system. To see if the Central European region (ceu) is installed on your system, for example:**

```
# localeadm -q ceu
locale/region name is ceu
Checking for Central Europe region (ceu)
.
.
.
The Central Europe region (ceu) is installed on this system
```

Changing System Information (Task Map)

Task	Directions	For Instructions
Manually set a system's date and time.	Manually set your system's date and time by using the `date` *mmddHHMM[[cc]yy]* command-line syntax.	"How to Set a System's Date and Time Manually" on page 72
Set up a message-of-the-day.	Set up a message-of-the-day on your system by editing the `/etc/motd` file.	"How to Set Up a Message-Of-The-Day" on page 72
Change a system's host name.	Change your system's host name by editing the following files: ■ `/etc/nodename` ■ `/etc/hostname.*`*host-name* ■ `/etc/inet/hosts` **Note –** If you are running the Solaris 3/05, 1/06, 6/06, or 11/06 releases, you also need to update the `/etc/inet/ipnodes` file. Starting with Solaris 10 8/07 release, the Solaris OS does not have two separate `hosts` files. The `/etc/inet/hosts` file is the single `hosts` file that contains both IPv4 and IPv6 entries.	"How to Change a System's Host Name" on page 73
Add a locale to a system.	Use the `localeadm` command to add a locale to your system.	How to Add a Locale to a System
Remove a locale from a system.	Use the `-r` option of the `localeadm` command and the locale to remove of locale from your system.	How to Remove a Locale From a System

Changing System Information

This section describes commands that enable you to change general system information.

▼ How to Set a System's Date and Time Manually

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Enter the new date and time.**

```
# date mmddHHMM[[cc]yy]
```

mm	Month, using two digits.
dd	Day of the month, using two digits.
HH	Hour, using two digits and a 24-hour clock.
MM	Minutes, using two digits.
cc	Century, using two digits.
yy	Year, using two digits.

See the `date(1)` man page for more information.

3 **Verify that you have reset your system's date correctly by using the `date` command with no options.**

Example 5–10 Setting a System's Date and Time Manually

The following example shows how to use the `date` command to manually set a system's date and time.

```
# date
Wed Mar  3 14:04:19 MST 2004
# date 0121173404
Thu Jan 21 17:34:34 MST 2004
```

▼ How to Set Up a Message-Of-The-Day

Edit the message-of-the-day file, `/etc/motd`, to include announcements or inquiries to all users of a system when they log in. Use this feature sparingly, and edit this file regularly to remove obsolete messages.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Edit the `/etc/motd` file and add a message of your choice.**

Edit the text to include the message that will be displayed during user login. Include spaces, tabs, and carriage returns.

3 **Verify the changes by displaying the contents of the `/etc/motd` file.**

```
$ cat /etc/motd
Welcome to the UNIX Universe. Have a nice day.
```

Example 5–11 Setting Up a Message-Of-The-Day

The default message-of-the-day, which is provided when you install Solaris software, contains SunOS version information.

```
$ cat /etc/motd
Sun Microsystems Inc.   SunOS 5.10       Generic  May 2004
```

The following example shows an edited `/etc/motd` file that provides information about system availability to each user who logs in.

```
$ cat /etc/motd
The system will be down from 7:00 a.m to 2:00 p.m. on
Saturday, July 7, for upgrades and maintenance.
Do not try to access the system during those hours.
Thank you.
```

▼ How to Change a System's Host Name

A system's host name is specified in several different locations.

Remember to update your name service database to reflect the new host name.

Use the following procedure to change or rename a system's host name.

You can also use the `sys-unconfig` command to reconfigure a system, including the host name. For more information, see the `sys-unconfig`(1M) man page.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Change the system's host name in the following files:**

- `/etc/nodename`
- `/etc/hostname.*`*interface*

- /etc/inet/hosts
- /etc/inet/ipnodes – Applies *only* to some Solaris releases.

Note – Starting with the Solaris 10 8/07 release, there is no longer two separate hosts files. The /etc/inet/hosts file is the single hosts file that contains both IPv4 and IPv6 entries. You do not need to maintain IPv4 entries in two hosts files that always require synchronization. For backward compatibility, the /etc/inet/ipnodes file is replaced with a symbolic link of the same name to the /etc/inet/hosts file. For more information, see the hosts(4) man page.

---

3 **(Optional) If you are using a name service, change the system's host name in the hosts file.**

4 **Rename the host name directory within the /var/crash directory.**

```
# cd /var/crash
# mv old-host-name  new-host-name
```

5 **Reboot the system to activate the new host name.**

```
# init 6
```

▼ How to Add a Locale to a System

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Add the packages for the locale you want to install on your system using the localeadm command. The -a option and a locale identifies the locale that you want to add. The -d option and a device identifies the device containing the locale packages you want to add. To add the Central European region (ceu) to your system, for example:**

```
# localeadm -a ceu -d /net/install/latest/Solaris/Product

locale/region name is ceu

Devices are /net/install/latest/Solaris/Product
.
.
.
One or more locales have been added.
To update the list of locales available at
.
.
.
```

▼ How to Remove a Locale From a System

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Remove the packages for the locale installed on your system using the `localeadm` command. The `-r` option and a locale identifies the locale that you want to remove from the system. To remove the Central European region (ceu) from your system, for example:**

   ```
   # localeadm -r ceu
   locale/region name is ceu
   Removing packages for Central Europe (ceu)
   .
   .
   .
   One or more locales have been removed.
   To update the list of locales available
   at the login screen's "Options->Language" menu,
   .
   .
   .
   ```

◆ ◆ ◆ CHAPTER 6

Managing Disk Use (Tasks)

This chapter describes how to optimize disk space by locating unused files and large directories.

For information on the procedures associated with managing disk use, see "Managing Disk Use (Task Map)" on page 77.

Managing Disk Use (Task Map)

Task	Description	For Instructions
Display information about files and disk space.	Display information about how disk space is used by using the `df` command.	"How to Display Information About Files and Disk Space" on page 79
Display the size of files.	Display information about the size of files by using the `ls` command with the `-lh` options.	"How to Display the Size of Files" on page 82
Find large files.	The `ls -s` command allows you to sort files by size, in descending order.	"How to Find Large Files" on page 83
Find files that exceed a specified size limit.	Locate and display the names of files that exceed a specified size by using the `find` command with the `-size` option and the value of the specified size limit.	"How to Find Files That Exceed a Specified Size Limit" on page 84
Display the size of directories, subdirectories, and files.	Display the size of one or more directories, subdirectories, and files by using the `du` command.	"How to Display the Size of Directories, Subdirectories, and Files" on page 85

Task	Description	For Instructions
Display ownership of local UFS file systems.	Display ownership of files by using the `quot -a` command.	“How to Display the User Ownership of Local UFS File Systems” on page 86
List the newest files.	Display the most recently created or changed files first, by using the `ls -t` command	“How to List the Newest Files” on page 88
Find and remove old or inactive files.	Use the `find` command with the `-atime` and `-mtime` options to locate files that have not been accessed for a specified number of days. You can remove these files by using the`rm ‘cat` *filename*`'` command.	“How to Find and Remove Old or Inactive Files” on page 89
Clear out temporary directories.	Locate temp directories, then use the `rm -r *` command to remove the entire directory.	“How to Clear Out Temporary Directories” on page 90
Find and delete core files.	Find and delete core files by using the `find . -name core -exec rm {} \;` command.	“How to Find and Delete `core` Files” on page 91
Delete crash dump files.	Delete crash dump files that are located in the `/var/crash/` directory by using the `rm *` command.	“How to Delete Crash Dump Files” on page 91

Displaying Information About Files and Disk Space

This table summarizes the commands available for displaying information about file size and disk space.

Command	Description	Man Page
`df`	Reports the number of free disk blocks and files	`df`(1M)
`du`	Summarizes disk space allocated to each subdirectory	`du`(1)
`find -size`	Searches recursively through a directory based on the size specified with the `-size` option	`find`(1)

Command	Description	Man Page
`ls -lh`	Lists the size of a file in the power of 1024 scaling	ls(1)

▼ How to Display Information About Files and Disk Space

- **Display information about how disk space is used by using the `df` command.**

```
$ df [directory]  [-h] [-t]
```

`df` With no options, lists all mounted file systems and their device names, the number of 512-byte blocks used, and the number of files.

directory Specifies the directory whose file system you want to check.

`-h` Displays disk space in the power of 1024 scaling.

`-t` Displays the total blocks as well as the blocks used for all mounted file systems.

Example 6–1 Displaying Information About File Size and Disk Space

In the following example, all the file systems listed are locally mounted except for `/usr/dist`.

```
$ df
/                   (/dev/dsk/c0t0d0s0 ):  101294 blocks   105480 files
/devices            (/devices          ):       0 blocks        0 files
/system/contract    (ctfs              ):       0 blocks 2147483578 files
/proc               (proc              ):       0 blocks     1871 files
/etc/mnttab         (mnttab            ):       0 blocks        0 files
/etc/svc/volatile   (swap              ):  992704 blocks    16964 files
/system/object      (objfs             ):       0 blocks 2147483530 files
/usr                (/dev/dsk/c0t0d0s6 ):  503774 blocks   299189 files
/dev/fd             (fd                ):       0 blocks        0 files
/var/run            (swap              ):  992704 blocks    16964 files
/tmp                (swap              ):  992704 blocks    16964 files
/opt                (/dev/dsk/c0t0d0s5 ):   23914 blocks     6947 files
/export/home        (/dev/dsk/c0t0d0s7 ):   16810 blocks     7160 files
```

Example 6–2 Displaying File Size Information in 1024 Bytes on a System With a UFS Root File System

In the following example, file system information for a system with a UFS root file system is displayed in 1024 bytes.

```
$ df -h
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c0t0d0s0      249M   200M    25M    90%    /
/devices                 0K     0K     0K     0%    /devices
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   485M   376K   485M     1%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
/dev/dsk/c0t0d0s6      3.2G   2.9G   214M    94%    /usr
fd                       0K     0K     0K     0%    /dev/fd
swap                   485M    40K   485M     1%    /var/run
swap                   485M    40K   485M     1%    /tmp
/dev/dsk/c0t0d0s5       13M   1.7M    10M    15%    /opt
/dev/dsk/c0t0d0s7      9.2M   1.0M   7.3M    13%    /export/home
```

Although /proc and /tmp are local file systems, they are not UFS file systems. /proc is a PROCFS file system, /var/run and /tmp are TMPFS file systems, and /etc/mnttab is an MNTFS file system.

Example 6–3 Displaying File Size Information in 1024 Bytes on a System With a ZFS Root File System

In the following example, file system information for a system with a ZFS root file system is displayed in 1024 bytes.

```
Filesystem             size   used  avail capacity  Mounted on
rpool/ROOT/s1008be      67G   4.6G    58G     8%    /
/devices                 0K     0K     0K     0%    /devices
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   1.9G   1.5M   1.9G     1%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
sharefs                  0K     0K     0K     0%    /etc/dfs/sharetab
/platform/sun4u-us3/lib/libc_psr/libc_psr_hwcap1.so.1
                        63G   4.6G    58G     8% /platform/sun4u-us3/lib/libc_psr.so.1
/platform/sun4u-us3/lib/sparcv9/libc_psr/libc_psr_hwcap1.so.1
                        63G   4.6G    58G     8% /platform/sun4u-us3/lib/sparcv9/libc_psr.so.1
fd                       0K     0K     0K     0%    /dev/fd
rpool/ROOT/s1008be/var
                        67G    73M    58G     1%    /var
swap                   1.9G    32K   1.9G     1%    /tmp
swap                   1.9G    40K   1.9G     1%    /var/run
rpool/export            67G    20K    58G     1%    /export
rpool/export/home       67G    18K    58G     1%    /export/home
```

Example 6–4 Displaying Total Number of Blocks and Files Allocated for a File System

The following example shows a list of all mounted file systems, device names, total 512-byte blocks used, and the number of files. The second line of each two-line entry displays the total number of blocks and files that are allocated for the file system.

```
$ df -t
/                  (/dev/dsk/c0t0d0s0 ):   101294 blocks   105480 files
                               total:   509932 blocks   129024 files
/devices           (/devices          ):        0 blocks        0 files
                               total:        0 blocks      113 files
/system/contract   (ctfs              ):        0 blocks 2147483578 files
                               total:        0 blocks       69 files
/proc              (proc              ):        0 blocks     1871 files
                               total:        0 blocks     1916 files
/etc/mnttab        (mnttab            ):        0 blocks        0 files
                               total:        0 blocks        1 files
/etc/svc/volatile  (swap              ):   992608 blocks    16964 files
                               total:   993360 blocks    17025 files
/system/object     (objfs             ):        0 blocks 2147483530 files
                               total:        0 blocks      117 files
/usr               (/dev/dsk/c0t0d0s6 ):   503774 blocks   299189 files
                               total:  6650604 blocks   420480 files
/dev/fd            (fd                ):        0 blocks        0 files
                               total:        0 blocks       31 files
/var/run           (swap              ):   992608 blocks    16964 files
                               total:   992688 blocks    17025 files
/tmp               (swap              ):   992608 blocks    16964 files
                               total:   992688 blocks    17025 files
/opt               (/dev/dsk/c0t0d0s5 ):    23914 blocks     6947 files
                               total:    27404 blocks     7168 files
/export/home       (/dev/dsk/c0t0d0s7 ):    16810 blocks     7160 files
                               total:    18900 blocks     7168 files
```

Checking the Size of Files

You can check the size of files and sort them by using the `ls` command. You can find files that exceed a size limit by using the `find` command. For more information, see the `ls(1)` and `find(1)` man pages.

Note – If you run out of space in the `/var` directory, do not symbolically link the `/var` directory to a directory on a file system with more disk space. Doing so, even as a temporary measure, might cause problems for certain Solaris daemon processes and utilities.

▼ How to Display the Size of Files

1. **Change to the directory where the files you want to check are located.**

2. **Display the size of the files.**

```
$ ls [-lh] [-s]
```

-l Displays a list of files and directories in long format, showing the sizes in bytes. (See the example that follows.)

-h Scales file sizes and directory sizes into Kbytes, Mbytes, Gbytes, or Tbytes when the file or directory size is larger than 1024 bytes. This option also modifies the output displayed by the -o, -n, -@, and -g options to display file or directory sizes in the new format. For more information, see the ls(1) man page.

-s Displays a list of the files and directories, showing the sizes in blocks.

Example 6–5 Displaying the Size of Files

The following example shows that the lastlog and messages files are larger than the other files in the /var/adm directory.

```
$ cd /var/adm
$ ls -lh
total 148
drwxrwxr-x   5 adm      adm          512 Nov 26 09:39 acct/
-rw-------   1 uucp     bin            0 Nov 26 09:25 aculog
drwxr-xr-x   2 adm      adm          512 Nov 26 09:25 exacct/
-r--r--r--   1 root     other       342K Nov 26 13:56 lastlog
drwxr-xr-x   2 adm      adm          512 Nov 26 09:25 log/
-rw-r--r--   1 root     root         20K Nov 26 13:55 messages
drwxr-xr-x   2 adm      adm          512 Nov 26 09:25 passwd/
drwxrwxr-x   2 adm      sys          512 Nov 26 09:39 sa/
drwxr-xr-x   2 root     sys          512 Nov 26 09:49 sm.bin/
-rw-rw-rw-   1 root     bin            0 Nov 26 09:25 spellhist
drwxr-xr-x   2 root     sys          512 Nov 26 09:25 streams/
-rw-r--r--   1 root     bin         3.3K Nov 26 13:56 utmpx
-rw-r--r--   1 root     root           0 Nov 26 10:17 vold.log
-rw-r--r--   1 adm      adm          19K Nov 26 13:56 wtmpx
```

The following example shows that the lpsched.1 file uses two blocks.

```
$ cd /var/lp/logs
$ ls -s
total 2            0 lpsched       2 lpsched.1
```

▼ How to Find Large Files

1. **Change to the directory that you want to search.**
2. **Display the size of files in blocks from largest to smallest.**
   - If the characters or columns for the files are *different*, use the following command to sort a list of files by block size, from largest to smallest.

     ```
     $ ls -l | sort +4rn | more
     ```

     Note that this command sorts files in a list by the character that is in the fourth field, starting from the left.
   - If the characters or columns for the files are the *same*, use the following command to sort a list of files by block size, from largest to smallest.

     ```
     $ ls -s | sort -nr | more
     ```

     Note that this command sorts files in a list, starting with the left most character.

Example 6–6 Finding Large Files (Sorting by the Fifth Field's Character)

```
$ cd /var/adm
$ ls -l | sort +4rn | more
-r--r--r--   1 root     root     4568368 Oct 17 08:36 lastlog
-rw-r--r--   1 adm      adm       697040 Oct 17 12:30 pacct.9
-rw-r--r--   1 adm      adm       280520 Oct 17 13:05 pacct.2
-rw-r--r--   1 adm      adm       277360 Oct 17 12:55 pacct.4
-rw-r--r--   1 adm      adm       264080 Oct 17 12:45 pacct.6
-rw-r--r--   1 adm      adm       255840 Oct 17 12:40 pacct.7
-rw-r--r--   1 adm      adm       254120 Oct 17 13:10 pacct.1
-rw-r--r--   1 adm      adm       250360 Oct 17 12:25 pacct.10
-rw-r--r--   1 adm      adm       248880 Oct 17 13:00 pacct.3
-rw-r--r--   1 adm      adm       247200 Oct 17 12:35 pacct.8
-rw-r--r--   1 adm      adm       246720 Oct 17 13:15 pacct.0
-rw-r--r--   1 adm      adm       245920 Oct 17 12:50 pacct.5
-rw-r--r--   1 root     root      190229 Oct  5 03:02 messages.1
-rw-r--r--   1 adm      adm       156800 Oct 17 13:17 pacct
-rw-r--r--   1 adm      adm       129084 Oct 17 08:36 wtmpx
```

Example 6–7 Finding Large Files (Sorting by the Left Most Character)

In the following example, the `lastlog` and `messages` files are the largest files in the `/var/adm` directory.

```
$ cd /var/adm
$ ls -s | sort -nr | more
  48 lastlog
  30 messages
  24 wtmpx
  18 pacct
   8 utmpx
   2 vold.log
   2 sulog
   2 sm.bin/
   2 sa/
   2 passwd/
   2 pacct1
   2 log/
   2 acct/
   0 spellhist
   0 aculog
total 144
```

▼ How to Find Files That Exceed a Specified Size Limit

- **To locate and display the names of files that exceed a specified size, use the `find` command.**

```
$ find directory -size +nnn
```

directory Identifies the directory that you want to search.

`-size +`*nnn* Is a number of 512-byte blocks. Files that exceed this size are listed.

Example 6–8 Finding Files That Exceed a Specified Size Limit

The following example shows how to find files larger than 400 blocks in the current working directory. The `-print` option displays the output of the `find` command.

```
$ find . -size +400 -print
./Howto/howto.doc
./Howto/howto.doc.backup
./Howto/howtotest.doc
./Routine/routineBackupconcepts.doc
./Routine/routineIntro.doc
./Routine/routineTroublefsck.doc
./.record
./Mail/pagination
./Config/configPrintadmin.doc
./Config/configPrintsetup.doc
./Config/configMailappx.doc
```

```
./Config/configMailconcepts.doc
./snapshot.rs
```

Checking the Size of Directories

You can display the size of directories by using the `du` command and options. Additionally, you can find the amount of disk space used by user accounts on local UFS file systems by using the `quot` command. For more information about these commands, see the `du(1)` and `quot(1M)` man pages.

▼ How to Display the Size of Directories, Subdirectories, and Files

- **Display the size of one or more directories, subdirectories, and files by using the `du` command. Sizes are displayed in 512-byte blocks.**

  $ **du [-as]** [*directory* ...]

  `du` Displays the size of each directory that you specify, including each subdirectory beneath it.

  `-a` Displays the size of each file and subdirectory, and the total number of blocks that are contained in the specified directory.

  `-s` Displays the total number of blocks that are contained in the specified directory.

  `-h` Displays the size of each directory in 1024–byte blocks.

  `-H` Displays the size of each directory in 1000–byte blocks.

  [*directory* ...] Identifies one or more directories that you want to check. Separate multiple directories in the command-line syntax with spaces.

Example 6–9 Displaying the Size of Directories, Subdirectories, and Files

The following example shows the sizes of two directories.

```
$ du -s /var/adm /var/spool/lp
130     /var/adm
40      /var/spool/lp
```

The following example shows the sizes of two directories and includes the sizes of all the subdirectories and files that are contained within each directory. The total number of blocks that are contained in each directory is also displayed.

```
$ du /var/adm /var/spool/lp
2       /var/adm/exacct
2       /var/adm/log
2       /var/adm/streams
2       /var/adm/acct/fiscal
2       /var/adm/acct/nite
2       /var/adm/acct/sum
8       /var/adm/acct
2       /var/adm/sa
2       /var/adm/sm.bin
258     /var/adm
4       /var/spool/lp/admins
2       /var/spool/lp/requests/printing.Eng.Sun.COM
4       /var/spool/lp/requests
4       /var/spool/lp/system
2       /var/spool/lp/fifos
24      /var/spool/lp
```

The following example shows directory sizes in 1024–byte blocks.

```
$ du -h /usr/share/audio
 796K   /usr/share/audio/samples/au
 797K   /usr/share/audio/samples
 798K   /usr/share/audio
```

How to Display the User Ownership of Local UFS File Systems

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Display users, directories, or file systems, and the number of 1024-byte blocks used.**

```
# quot [-a] [filesystem ...]
```

`-a` Lists all users of each mounted UFS file system and the number of 1024-byte blocks used.

filesystem Identifies a UFS file system. Users and the number of blocks used are displayed for that file system.

Note – The quot command works only on local UFS file systems.

Example 6–10 Displaying the User Ownership of Local UFS File Systems

In the following example, users of the root (/) file system are displayed. In the subsequent example, users of all mounted UFS file systems are displayed.

```
# quot /
/dev/rdsk/c0t0d0s0:
43340   root
 3142   rimmer
   47   uucp
   35   lp
   30   adm
    4   bin
    4   daemon

# quot -a
/dev/rdsk/c0t0d0s0 (/):
43340   root
 3150   rimmer
   47   uucp
   35   lp
   30   adm
    4   bin
    4   daemon
/dev/rdsk/c0t0d0s6 (/usr):
460651  root
206632  bin
  791   uucp
   46   lp
    4   daemon
    1   adm
/dev/rdsk/c0t0d0s7 (/export/home):
    9   root
```

Finding and Removing Old or Inactive Files

Part of the job of cleaning up heavily loaded file systems involves locating and removing files that have not been used recently. You can locate unused files by using the ls or find commands. For more information, see the ls(1) and find(1) man pages.

Other ways to conserve disk space include emptying temporary directories such as the directories located in `/var/tmp` or `/var/spool`, and deleting `core` and crash dump files. For more information about crash dump files, refer to Chapter 17, "Managing System Crash Information (Tasks)."

How to List the Newest Files

- **List files, displaying the most recently created or changed files first, by using the `ls -t` command.**

```
$ ls -t [directory]
```

`-t` Sorts files by latest time stamp first.

directory Identifies the directory that you want to search.

Example 6–11 Listing the Newest Files

The following example shows how to use the `ls -tl` command to locate the most recently created or changed files within the `/var/adm` directory. The `sulog` file was created or edited most recently.

```
$ ls -tl /var/adm
total 134
-rw-------   1 root     root         315 Sep 24 14:00 sulog
-r--r--r--   1 root     other     350700 Sep 22 11:04 lastlog
-rw-r--r--   1 root     bin         4464 Sep 22 11:04 utmpx
-rw-r--r--   1 adm      adm        20088 Sep 22 11:04 wtmpx
-rw-r--r--   1 root     other          0 Sep 19 03:10 messages
-rw-r--r--   1 root     other          0 Sep 12 03:10 messages.0
-rw-r--r--   1 root     root       11510 Sep 10 16:13 messages.1
-rw-r--r--   1 root     root           0 Sep 10 16:12 vold.log
drwxr-xr-x   2 root     sys          512 Sep 10 15:33 sm.bin
drwxrwxr-x   5 adm      adm          512 Sep 10 15:19 acct
drwxrwxr-x   2 adm      sys          512 Sep 10 15:19 sa
-rw-------   1 uucp     bin            0 Sep 10 15:17 aculog
-rw-rw-rw-   1 root     bin            0 Sep 10 15:17 spellhist
drwxr-xr-x   2 adm      adm          512 Sep 10 15:17 log
drwxr-xr-x   2 adm      adm          512 Sep 10 15:17 passwd
```

▼ How to Find and Remove Old or Inactive Files

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see “Configuring RBAC (Task Map)” in *System Administration Guide: Security Services*.

2 **Find files that have not been accessed for a specified number of days and list them in a file.**

```
# find directory -type f[-atime +nnn] [-mtime +nnn] -print > filename &
```

directory Identifies the directory you want to search. Directories below this directory are also searched.

`-atime +`*nnn* Finds files that have not been accessed within the number of days (nnn) that you specify.

`-mtime +`*nnn* Finds files that have not been modified within the number of days (nnn) that you specify.

filename Identifies the file that contains the list of inactive files.

3 **Remove the inactive files found listed in the previous step.**

```
# rm ‘cat filename‘
```

where *filename* identifies the file that was created in the previous step. This file contains the list of inactive files.

Example 6–12 Finding and Removing Old or Inactive Files

The following example shows files in the `/var/adm` directory and the subdirectories that have not been accessed in the last 60 days. The `/var/tmp/deadfiles` file contains the list of inactive files. The `rm` command removes these inactive files.

```
# find /var/adm -type f -atime +60 -print > /var/tmp/deadfiles &
# more /var/tmp/deadfiles
/var/adm/aculog
/var/adm/spellhist
/var/adm/wtmpx
/var/adm/sa/sa13
/var/adm/sa/sa27
/var/adm/sa/sa11
/var/adm/sa/sa23
/var/adm/sulog
/var/adm/vold.log
/var/adm/messages.1
/var/adm/messages.2
/var/adm/messages.3
```

```
# rm ‘cat /var/tmp/deadfiles‘
#
```

▼ How to Clear Out Temporary Directories

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Change to the directory that you want to clean out.**

```
# cd directory
```

Caution – Ensure that you are in the correct directory before completing Step 3. Step 3 deletes all files in the current directory.

3 **Delete the files and subdirectories in the current directory.**

```
# rm -r *
```

4 **Change to other directories that contain unnecessary, temporary or obsolete subdirectories and files. Delete these subdirectories and files by repeating Step 3.**

Example 6–13 Clearing Out Temporary Directories

The following example shows how to clear out the mywork directory, and how to verify that all files and subdirectories were removed.

```
# cd mywork
# ls
filea.000
fileb.000
filec.001
# rm -r *
# ls
#
```

▼ How to Find and Delete `core` Files

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Change to the directory where you want to search for `core` files.**

3 **Find and remove any `core` files in this directory and its subdirectories.**

```
# find . -name core -exec rm {} \;
```

Example 6–14 Finding and Deleting `core` Files

The following example shows how to find and remove `core` files from the `jones` user account by using the `find` command.

```
# cd /home/jones
# find . -name core -exec rm {} \;
```

▼ How to Delete Crash Dump Files

Crash dump files can be very large. If you have enabled your system to store these files, do not retain them for longer than necessary.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Change to the directory where crash dump files are stored.**

```
# cd /var/crash/system
```

where *system* identifies a system that created the crash dump files.

⚠ **Caution –** Ensure you are in the correct directory before completing Step 3. Step 3 deletes all files in the current directory.

3 **Remove the crash dump files.**

```
# rm *
```

4 **Verify that the crash dump files were removed.**

```
# ls
```

Example 6–15 Deleting Crash Dump Files

The following example shows how to remove crash dump files from the system venus, and how to verify that the crash dump files were removed.

```
# cd /var/crash/venus
# rm *
# ls
```

◆ ◆ ◆ CHAPTER 7

Managing UFS Quotas (Tasks)

This chapter describes how to set up and administer UFS quotas for disk space and inodes.

For information associated with managing UFS quotas, see the following:

- "Setting Up Quotas (Task Map)" on page 96
- "Maintaining Quotas (Task Map)" on page 100

For information about managing ZFS quotas, see "Setting ZFS Quotas and Reservations" in *Solaris ZFS Administration Guide*.

What Are Quotas?

Quotas enable system administrators to control the size of file systems. Quotas limit the amount of disk space and the number of inodes, which roughly corresponds to the number of files, that individual users can acquire. For this reason, quotas are especially useful on the file systems where user home directories reside. As a rule, the `public` and `/tmp` file systems usually do not benefit significantly by establishing quotas.

Using Quotas

Once quotas are in place, they can be changed to adjust the amount of disk space or the number of inodes that users can consume. Additionally, quotas can be added or removed as system needs change. For instructions on changing quotas or the amount of time that quotas can be exceeded, disabling individual quotas, or removing quotas from file systems, see "Changing and Removing Quotas" on page 103.

In addition, quota status can be monitored. Quota commands enable administrators to display information about quotas on a file system, or search for users who have exceeded their quotas. For procedures that describe how to use these commands, see "Checking Quotas" on page 101.

Setting Soft Limits and Hard Limits for Quotas

You can set both soft limits and hard limits. The system does not allow a user to exceed his or her hard limit. However, a system administrator might set a soft limit, which the user can temporarily exceed. The soft limit must be less than the hard limit.

Once the user exceeds the soft limit, a quota timer begins. While the quota timer is ticking, the user is allowed to operate above the soft limit but cannot exceed the hard limit. Once the user goes below the soft limit, the timer is reset. However, if the user's usage remains above the soft limit when the timer expires, the soft limit is enforced as a hard limit. By default, the soft limit timer is set to seven days.

The `timeleft` field in the `repquota` and `quota` commands shows the value of the timer.

For example, let's say a user has a soft limit of 10,000 blocks and a hard limit of 12,000 blocks. If the user's block usage exceeds 10,000 blocks and the seven-day timer is also exceeded, the user cannot allocate more disk blocks on that file system until his or her usage drops below the soft limit.

The Difference Between Disk Block and File Limits

A file system provides two resources to the user, blocks for data and inodes for files. Each file consumes one inode. File data is stored in data blocks. Data blocks are usually made up of 1Kbyte blocks.

Assuming no directories exist, a user can exceed his or her inode quota by creating all empty files without using any blocks. A user can also use one inode, yet exceed his or her block quota, by creating one file that is large enough to consume all the data blocks in the user's quota.

Setting Up Quotas

Setting up quotas involves these general steps:

1. Ensuring that quotas are enforced each time the system is rebooted by adding a quota option to the `/etc/vfstab` file entries. Also, creating a `quotas` file in the top-level directory of the file system.
2. After you create a quota for one use, copying the quota as a prototype to set up other user quotas.
3. Before you turn quotas on, checking the consistency of the proposed quotas with the current disk usage to make sure that there are no conflicts.
4. Turning on the quotas on for one or more file systems.

For specific information about these procedures, see “Setting Up Quotas (Task Map)” on page 96.

The following table describes the commands that you use to set up disk quotas.

TABLE 7–1 Commands for Setting Up Quotas

Command	Task	Man Page
`edquota`	Sets the hard limits and soft limits on the number of inodes and the amount of disk space for each user.	`edquota`(1M)
`quotacheck`	Examines each mounted UFS file system, comparing the file system's current disk usage against information stored in the file system's disk quota file. Then, resolves inconsistencies	`quotacheck`(1M)
`quotaon`	Activates the quotas for the specified file systems.	`quotaon`(1M)
`quota`	Displays users' disk quotas on mounted file systems to verify that the quotas have been correctly set up.	`quota`(1M)

Guidelines for Setting Up Quotas

Before you set up quotas, you need to determine how much disk space and how many inodes to allocate to each user. If you want to ensure that the total file system space is never exceeded, you can divide the total size of the file system between the number of users. For example, if three users share a 100-Mbyte slice and have equal disk space needs, you could allocate 33 Mbytes to each user.

In environments where not all users are likely to push their limits, you might want to set individual quotas so that they add up to more than the total size of the file system. For example, if three users share a 100-Mbyte slice, you could allocate 40 Mbytes to each user.

When you have established a quota for one user by using the `edquota` command, you can use this quota as a prototype to set the same quota for other users on the same file system.

Before you turn on the quotas, do the following:

- First, configure the UFS file systems for the quotas.
- Establish quotas for each user, and run the `quotacheck` command to check for consistency between current disk usage and quota files.
- Run the `quotacheck` command periodically if systems are rebooted infrequently.

The quotas you set up with the `edquota` command are not enforced until you turn them on by using the `quotaon` command. If you have properly configured the quota files, the quotas are turned on automatically each time a system is rebooted and the file system is mounted.

Setting Up Quotas (Task Map)

Task	Description	For Instructions
1. Configure a file system for quotas.	Edit the /etc/vfstab file so that quotas are activated each time the file system is mounted. Also, create a quotas file.	"How to Configure File Systems for Quotas" on page 96
2. Set up quotas for a user.	Use the edquota command to create disk quotas and inode quotas for a single user account.	"How to Set Up Quotas for a User" on page 97
3. (Optional) Set up quotas for multiple users.	Use the edquota command to apply prototype quotas to other user accounts.	"How to Set Up Quotas for Multiple Users" on page 98
4. Check for consistency.	Use the quotacheck command to compare quotas to current disk usage for consistency across one or more file systems.	"How to Check Quota Consistency" on page 98
5. Turn on quotas.	Use the quotaon command to initiate quotas on one or more file systems.	"How to Turn On Quotas" on page 99

▼ How to Configure File Systems for Quotas

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Edit the /etc/vfstab file and add rq to the mount options field for each UFS file system that will have quotas.**

3 **Change directory to the root of the file system that will have quotas.**

4 **Create a file named quotas.**

```
# touch quotas
```

5 **Change permissions to read/write for superuser access only.**

```
# chmod 600 quotas
```

Example 7–1 Configuring File Systems for Quotas

The following /etc/vfstab example shows that the /export/home directory from the system pluto is mounted as an NFS file system on the local system. You can tell that quotas are enabled by the rq entry under the mount options column.

```
# device    device    mount     FS    fsck   mount   mount
# to mount       to fsck  point     type   pass  at boot options
# pluto:/export/home -   /export/home nfs    -     yes    rq
```

The following example line from the /etc/vfstab file shows that the local /work directory is mounted with quotas enabled, signified by the rq entry under the mount options column.

```
#device     device        mount  FS   fsck  mount    mount
#to mount         to fsck      point  type  pass  at boot options
#/dev/dsk/c0t4d0s0 /dev/rdsk/c0t4d0s0 /work ufs  3    yes     rq
```

See Also

- "How to Set Up Quotas for a User" on page 97
- "How to Set Up Quotas for Multiple Users" on page 98
- "How to Check Quota Consistency" on page 98
- "How to Turn On Quotas" on page 99

▼ How to Set Up Quotas for a User

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Use the quota editor to create a temporary file that contains one line of quota information for each mounted UFS file system that has a quotas file in the file system's root directory.**

   ```
   # edquota username
   ```

   where *username* is the user for whom you want to set up quotas.

3. **Change the number of 1-Kbyte disk blocks, both soft and hard, and the number of inodes, both soft and hard, from the default of 0, to the quotas that you specify for each file system.**

4. **Verify the user's quota.**

   ```
   # quota -v username
   ```

   -v Displays the user's quota information on all mounted file systems where quotas exist.

   username Specifies the user name to view quota limits.

Example 7–2 Setting Up Quotas for a User

The following example shows the contents of the temporary file opened by `edquota` on a system where `/files` is the only mounted file system that contains a `quotas` file in the root directory.

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

The following example shows the same line in the temporary file after quotas have been set up.

```
fs /files blocks (soft = 50, hard = 60) inodes (soft = 90, hard = 100)
```

▼ How to Set Up Quotas for Multiple Users

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Use the quota editor to apply the quotas you already established for a prototype user to the additional users that you specify.**

   `# edquota -p` *prototype-user username ...*

   prototype-user Is the user name of the account for which you have set up quotas.

   username ... Specifies one or more user names of additional accounts. More than one user name is specified by separating each user name with a space.

Example 7–3 Setting Up Prototype Quotas for Multiple Users

The following example shows how to apply the quotas established for user `bob` to users `mary` and `john`.

```
# edquota -p bob mary john
```

▼ How to Check Quota Consistency

The `quotacheck` command is run automatically when a system is rebooted. You generally do not have to run the `quotacheck` command on an empty file system with quotas. However, if you are setting up quotas on a file system with existing files, you need to run the `quotacheck` command to synchronize the quota database with the files or inodes that already exist in the file system.

Also keep in mind that running the quotacheck command on large file systems can be time-consuming.

Note – To ensure accurate disk data, the file systems being checked should be quiescent when you run the quotacheck command manually.

---

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Run a consistency check on UFS file systems.**

```
# quotacheck [-va] filesystem
```

-v (Optional) Identifies the disk quotas for each user on a particular file system.

-a Checks all file systems with an rq entry in the /etc/vfstab file.

filesystem Specifies the file system to check.

See the quotacheck(1M) man page for more information.

Example 7–4 Checking Quota Consistency

The following example shows how to check quotas for the /export/home file system on the /dev/rdsk/c0t0d0s7 slice. The /export/home file system is the only file system with an rq entry in the /etc/vfstab file.

```
# quotacheck -va
*** Checking quotas for /dev/rdsk/c0t0d0s7 (/export/home)
```

▼ How to Turn On Quotas

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Turn on file system quotas.**

```
# quotaon [-v] -a filesystem ...
```

-v Displays a message for each file system after quotas are turned on.

-a Turns on quotas for all file systems with an rq entry in the /etc/vfstab file.

filesystem ... Turns on quotas for one or more file systems that you specify. More than one file system is specified by separating each file system name with a space.

Example 7–5 Turning On Quotas

The following example shows how to turn quotas on for the file systems on the /dev/dsk/c0t4d0s7 and /dev/dsk/c0t3d0s7 slices.

```
# quotaon -v /dev/dsk/c0t4d0s7 /dev/dsk/c0t3d0s7
/dev/dsk/c0t4d0s7: quotas turned on
/dev/dsk/c0t3d0s7: quotas turned on
```

Maintaining Quotas (Task Map)

Task	Description	For Instructions
Check for exceeded quotas.	Display the quotas and disk use for individual users on file systems on which quotas have been activated by using the quota command.	"How to Check for Exceeded Quotas" on page 101
Check for quotas on a file system.	Display the quotas and disk use for all users on one or more file systems by using the repquota command.	"How to Check Quotas on a File System" on page 102
Change the soft limit default.	Change the length of time that users can exceed their disk space quotas or inode quotas by using the edquota command.	"How to Change the Soft Limit Default" on page 103
Change quotas for a user.	Use the quota editor, edquota, to change quotas for an individual user.	"How to Change Quotas for a User" on page 104
Disable quotas for a user.	Use the quota editor, edquota, to disable quotas for an individual user.	"How to Disable Quotas for a User" on page 105
Turn off quotas.	Turn off quotas by using the quotaoff command.	"How to Turn Off Quotas" on page 106

Checking Quotas

After you have set up and turned on disk quotas and inode quotas, you can check for users who exceed their quotas. In addition, you can check quota information for entire file systems.

The following table describes the commands that you use to check quotas.

TABLE 7–2 Commands for Checking Quotas

Command	Task
`quota`(1M)	Displays user quotas and current disk use, and information about users who are exceeding their quotas
`repquota`(1M)	Displays quotas, files, and the amount of space that is owned for specified file systems

How to Check for Exceeded Quotas

You can display the quotas and disk use for individual users on file systems on which quotas have been activated by using the `quota` command.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Display user quotas for mounted file systems where quotas are enabled.**

```
# quota [-v] username
```

`-v` Displays one or more users' quotas on all mounted file systems that have quotas.

username Is the login name or UID of a user's account.

Example 7–6 Checking for Exceeded Quotas

The following example shows that the user account identified by UID 301 has one 1–Kbyte quota but has not used any disk space.

```
# quota -v 301
Disk quotas for bob (uid 301):
Filesystem  usage  quota limit timeleft files quota  limit timeleft
/export/home   0      1     2              0      2      3
```

`Filesystem` Is the mount point for the file system.

`usage` Is the current block usage.

`quota` Is the soft-block limit.

`limit` Is the hard-block limit.

`timeleft` Is the amount of time, in days, left on the quota timer.

`files` Is the current inode usage.

`quota` Is the soft-inode limit.

`limit` Is the hard-inode limit.

`timeleft` Is the amount of time, in days, left on the quota timer.

▼ How to Check Quotas on a File System

Display the quotas and disk use for all users on one or more file systems by using the `repquota` command.

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Display all quotas for one or more file systems, even if there is no usage.**

   ```
   # repquota [-v]  -a filesystem
   ```

   `-v` Reports on quotas for all users, even those users who do not consume resources.

   `-a` Reports on all file systems.

   filesystem Reports on the specified file system.

Example 7–7 Checking Quotas on a File System

The following example shows output from the `repquota` command on a system that has quotas enabled on only one file system (`/export/home`).

```
# repquota -va
/dev/dsk/c0t3d0s7 (/export/home):
                  Block limits              File limits
User      used   soft   hard  timeleft  used  soft   hard  timeleft
#301  --            0      1   2.0 days         0      2      3
#341  --    57     50     60   7.0 days  2      90    100
```

`Block limits` Definition

`used` Is the current block usage.

`soft`	Is the soft-block limit.
`hard`	Is the hard-block limit.
`timeleft`	Is the amount of time, in days, left on the quota timer.
`File limits`	Definition
`used`	Is the current inode usage.
`soft`	Is the soft-inode limit.
`hard`	Is the hard-inode limit.
`timeleft`	Is the amount of time, in days, left on the quota timer.

Changing and Removing Quotas

You can change quotas to adjust the amount of disk space or the number of inodes that users can consume. You can also remove quotas, for individual users or from entire file systems, as needed.

The following table describes the commands that you use to change quotas or to remove quotas.

TABLE 7–3 Commands for Changing Quotas and Removing Quotas

Command	Man Page	Description
`edquota`	`edquota`(1M)	Changes the hard limits and soft limits on the number of inodes or amount of disk space for each user. Also, changes the soft limit for each file system with a quota.
`quotaoff`	`quotaon`(1M)	Turns off quotas for specified file systems.

How to Change the Soft Limit Default

By default, users can exceed the soft time limits for their quotas for one week. So, after a week of repeated violations of the soft time limits of either disk space quotas or inode quotas, the system prevents users from using any more inodes or disk blocks.

You can change the length of time that users can exceed their disk space quotas or inode quotas by using the `edquota` command.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Use the quota editor to create a temporary file that contains soft time limits.**

```
# edquota -t
```

where the `-t` option specifies the editing of the soft time limits for each file system.

3 **Change the time limits from 0 (the default) to the time limits that you specify. So, use numbers and the keywords `month`, `week`, `day`, `hour`, `min`, or `sec`.**

Note – This procedure does not affect current quota violators.

Example 7–8 Changing the Soft Limit Default

The following example shows the contents of the temporary file opened by the `edquota` command on a system where `/export/home` is the only mounted file system with quotas. The default value, `0`, means that the default time limit of one week is used.

```
fs /export/home blocks time limit = 0 (default), files time limit = 0 (default)
```

The following example shows the same temporary file after the time limit for exceeding the blocks quota has been changed to 2 weeks. Also, the time limit for exceeding the number of files has been changed to 16 days.

```
fs /export/home blocks time limit = 2 weeks, files time limit = 16 days
```

▼ How to Change Quotas for a User

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Use the quota editor to open a temporary file that contains one line for each mounted file system that has a `quotas` file in the file system's root directory.**

```
# edquota username
```

where *username* specifies the user name whose quota you want to change.

Caution – You can specify multiple users as arguments to the `edquota` command. However, the user that this information belongs to, is not displayed. To avoid confusion, specify only one user name.

3 **Specify the number of 1-Kbyte disk blocks, both soft and hard, and the number of inodes, both soft and hard.**

4 **Verify that a user's quota has been correctly changed.**

```
# quota -v username
```

-v Displays user quota information on all mounted file systems with quotas enabled.

username Specifies the user name whose quota you want to check.

Example 7–9 Changing Quotas for a User

The following example shows the contents of the temporary file opened by the `edquota` command. This temporary file is opened on a system where `/files` is the only mounted file system containing a `quotas` file in the file system's root directory.

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

The following output shows the same temporary file after quotas have been changed.

```
fs /files blocks (soft = 0, hard = 500) inodes (soft = 0, hard = 100)
```

Example 7–10 Verifying That Hard Quotas Have Been Changed

The following example shows how to verify that the hard quotas for user `smith` have been changed to 500 1-Kbyte blocks, and 100 inodes.

```
# quota -v smith
Disk quotas for smith (uid 12):
Filesystem  usage  quota  limit  timeleft  files  quota  limit  timeleft

  /files     1       0     500              1       0     100
```

▼ How to Disable Quotas for a User

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Use the quota editor to create a temporary file containing one line for each mounted file system that has a `quotas` file in its top-level directory.**

```
# edquota username
```

Where *username* specifies the user name whose quota you want to disable.

Caution – You can specify multiple users as arguments to the `edquota` command. However, the user that this information belongs to, is not displayed. To avoid confusion, specify only one user name.

3 **Change the number of 1-Kbyte disk blocks, both soft and hard, and the number of inodes, both soft and hard, to 0.**

Note – Ensure that you change the values to zero. Do *not* delete the line from the text file.

4 **Verify that you have disabled a user's quota.**

```
# quota -v username
```

`-v` Displays user quota information on all mounted file systems with quotas enabled.

username Specifies the user name (UID) whose quota you want to check.

Example 7–11 Disabling Quotas for a User

The following example shows the contents of the temporary file opened by the `edquota` command on a system where `/files` is the only mounted file system that contains a quotas file in the file system's root directory.

```
fs /files blocks (soft = 50, hard = 60) inodes (soft = 90, hard = 100)
```

The following example shows the same temporary file after quotas have been disabled.

```
fs /files blocks (soft = 0, hard = 0) inodes (soft = 0, hard = 0)
```

▼ How to Turn Off Quotas

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Turn off file system quotas.**

```
# quotaoff [-v] -a filesystem ...
```

`-v` Displays a message from each file system when quotas are turned off.

`-a` Turns off quotas for all file systems.

filesystem Turns off quotas for one or more file systems that you specify. More than one file system is specified by separating each file system name with a space.

Example 7–12 Turning Off Quotas

The following example shows how to turn off the quotas for the `/export/home` file system.

```
# quotaoff -v /export/home
/export/home: quotas turned off
```

CHAPTER 8

Scheduling System Tasks (Tasks)

This chapter describes how to schedule routine or single (one-time) system tasks by using the `crontab` and `at` commands.

This chapter also explains how to control access to these commands by using the following files:

- `cron.deny`
- `cron-allow`
- `at.deny`

For information on the procedures that are associated with scheduling system tasks, see the following:

- "Creating and Editing `crontab` Files (Task Map)" on page 109
- "Using the `at` Command (Task Map)" on page 122

Creating and Editing `crontab` Files (Task Map)

Task	Description	For Instructions
Create or edit a `crontab` file.	Use the `crontab -e` command to create or edit a `crontab` file.	"How to Create or Edit a `crontab` File" on page 115
Verify that a `crontab` file exists.	Use the `ls -l` command to verify the contents of the `/var/spool/cron/crontabs` file.	"How to Verify That a `crontab` File Exists" on page 116
Display a `crontab`file.	Use the `ls -l` command to display the `crontab` file.	"How to Display a `crontab` File" on page 117

Task	Description	For Instructions
Remove a crontab file	The crontab file is set up with restrictive permissions Use the crontab -r command, rather than the rm command to remove a crontab file.	"How to Remove a crontab File" on page 118
Deny crontab access	To deny users access to crontab commands, add user names to the /etc/cron.d/cron.deny file by editing this file.	"How to Deny crontab Command Access" on page 120
Limit crontab access to specified users.	To allow users access to the crontab command, add user names to the /etc/cron.d/cron.allow file.	"How to Limit crontab Command Access to Specified Users" on page 120

Ways to Automatically Execute System Tasks

You can set up many system tasks to execute automatically. Some of these tasks should occur at regular intervals. Other tasks need to run only once, perhaps during off hours such as evenings or weekends.

This section contains overview information about two commands, crontab and at, which enable you to schedule routine tasks to execute automatically. The crontab command schedules repetitive commands. The at command schedules tasks that execute once.

The following table summarizes crontab and at commands, as well as the files that enable you to control access to these commands.

TABLE 8–1 Command Summary: Scheduling System Tasks

Command	What It Schedules	Location of Files	Files That Control Access
crontab	Multiple system tasks at regular intervals	/var/spool/cron/crontabs	/etc/cron.d/cron.allow and /etc/cron.d/cron.deny
at	A single system task	/var/spool/cron/atjobs	/etc/cron.d/at.deny

You can also use the Solaris Management Console's Scheduled Jobs tool to schedule routine tasks. For information on using and starting the Solaris Management Console, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

For Scheduling Repetitive Jobs: crontab

You can schedule routine system administration tasks to execute daily, weekly, or monthly by using the crontab command.

Daily crontab system administration tasks might include the following:

- Removing files more than a few days old from temporary directories
- Executing accounting summary commands
- Taking snapshots of the system by using the df and ps commands
- Performing daily security monitoring
- Running system backups

Weekly crontab system administration tasks might include the following:

- Rebuilding the catman database for use by the man -k command
- Running the fsck -n command to list any disk problems

Monthly crontab system administration tasks might include the following:

- Listing files not used during a specific month
- Producing monthly accounting reports

Additionally, users can schedule crontab commands to execute other routine system tasks, such as sending reminders and removing backup files.

For step-by-step instructions on scheduling crontab jobs, see "How to Create or Edit a crontab File" on page 115.

For Scheduling a Single Job: at

The at command allows you to schedule a job for execution at a later time. The job can consist of a single command or a script.

Similar to crontab, the at command allows you to schedule the automatic execution of routine tasks. However, unlike crontab files, at files execute their tasks once. Then, they are removed from their directory. Therefore, the at command is most useful for running simple commands or scripts that direct output into separate files for later examination.

Submitting an at job involves typing a command and following the at command syntax to specify options to schedule the time your job will be executed. For more information about submitting at jobs, see "Description of the at Command" on page 123.

The at command stores the command or script you ran, along with a copy of your current environment variable, in the /var/spool/cron/atjobs directory. Your at job file name is given a long number that specifies its location in the at queue, followed by the .a extension, such as 793962000.a.

The cron daemon checks for at jobs at startup and listens for new jobs that are submitted. After the cron daemon executes an at job, the at job's file is removed from the atjobs directory. For more information, see the at(1) man page.

For step-by-step instructions on scheduling at jobs, see "How to Create an at Job" on page 124.

Scheduling a Repetitive System Task (cron)

The following sections describe how to create, edit, display, and remove crontab files, as well as how to control access to them.

Inside a crontab File

The cron daemon schedules system tasks according to commands found within each crontab file. A crontab file consists of commands, one command per line, that will be executed at regular intervals. The beginning of each line contains date and time information that tells the cron daemon when to execute the command.

For example, a crontab file named root is supplied during SunOS software installation. The file's contents include these command lines:

```
10 3 * * * /usr/sbin/logadm     (1)
15 3 * * 0 /usr/lib/fs/nfs/nfsfind     (2)
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1     (3)
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean     (4)
```

The following describes the output for each of these command lines:

- The first line runs the logadm command at 3:10 a.m. every day.
- The second line executes the nfsfind script every Sunday at 3:15 a.m.
- The third line runs a script that checks for daylight savings time (and make corrections, if necessary) at 2:10 a.m. daily.

  If there is no RTC time zone, nor an /etc/rtc_config file, this entry does nothing.

  x86 only – The /usr/sbin/rtc script can only be run on an x86 based system.

- The fourth line checks for (and removes) duplicate entries in the Generic Security Service table, /etc/gss/gsscred_db, at 3:30 a.m. daily.

For more information about the syntax of lines within a crontab file, see "Syntax of crontab File Entries" on page 114.

The crontab files are stored in the /var/spool/cron/crontabs directory. Several crontab files besides root are provided during SunOS software installation. See the following table.

TABLE 8–2 Default crontab Files

crontab File	Function
adm	Accounting
lp	Printing
root	General system functions and file system cleanup
sys	Performance data collection
uucp	General uucp cleanup

Besides the default crontab files, users can create crontab files to schedule their own system tasks. Other crontab files are named after the user accounts in which they are created, such as bob, mary, smith, or jones.

To access crontab files that belong to root or other users, superuser privileges are required.

Procedures explaining how to create, edit, display, and remove crontab files are described in subsequent sections.

How the cron Daemon Handles Scheduling

The cron daemon manages the automatic scheduling of crontab commands. The role of the cron daemon is to check the /var/spool/cron/crontab directory for the presence of crontab files.

The cron daemon performs the following tasks at startup:

- Checks for new crontab files.
- Reads the execution times that are listed within the files.
- Submits the commands for execution at the proper times.
- Listens for notifications from the crontab commands regarding updated crontab files.

In much the same way, the cron daemon controls the scheduling of at files. These files are stored in the /var/spool/cron/atjobs directory. The cron daemon also listens for notifications from the crontab commands regarding submitted at jobs.

Syntax of crontab File Entries

A crontab file consists of commands, one command per line, that execute automatically at the time specified by the first five fields of each command line. These five fields, described in the following table, are separated by spaces.

TABLE 8–3 Acceptable Values for crontab Time Fields

Time Field	Values
Minute	0-59
Hour	0-23
Day of month	1-31
Month	1-12
Day of week	0-6 (0 = Sunday)

Follow these guidelines for using special characters in crontab time fields:

- Use a space to separate each field.
- Use a comma to separate multiple values.
- Use a hyphen to designate a range of values.
- Use an asterisk as a wildcard to include all possible values.
- Use a comment mark (#) at the beginning of a line to indicate a comment or a blank line.

For example, the following crontab command entry displays a reminder in the user's console window at 4 p.m. on the first and fifteenth days of every month.

```
0 16 1,15 * * echo Timesheets Due > /dev/console
```

Each command within a crontab file must consist of one line, even if that line is very long. The crontab file does not recognize extra carriage returns. For more detailed information about crontab entries and command options, refer to the crontab(1) man page.

Creating and Editing crontab Files

The simplest way to create a crontab file is to use the crontab -e command. This command invokes the text editor that has been set for your system environment. The default editor for your system environment is defined in the EDITOR environment variable. If this variable has not been set, the crontab command uses the default editor, ed. Preferably, you should choose an editor that you know well.

The following example shows how to determine if an editor has been defined, and how to set up vi as the default.

```
$ which $EDITOR
$
$ EDITOR=vi
$ export EDITOR
```

When you create a crontab file, it is automatically placed in the /var/spool/cron/crontabs directory and is given your user name. You can create or edit a crontab file for another user, or root, if you have superuser privileges.

▼ How to Create or Edit a crontab File

Before You Begin If you are creating or editing a crontab file that belongs to root or another user you must become superuser or assume an equivalent role. Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*

You do not need to become superuser to edit your own crontabfile.

1 **Create a new crontab file, or edit an existing file.**

```
$ crontab -e [username]
```

where *username* specifies the name of the user's account for which you want to create or edit a crontab file. You can create your own crontab file without superuser privileges, but you must have superuser privileges to creating or edit a crontab file for root or another user.

Caution – If you accidentally type the crontab command with no option, press the interrupt character for your editor. This character allows you to quit without saving changes. If you instead saved changes and exited the file, the existing crontab file would be overwritten with an empty file.

---

2 **Add command lines to the crontab file.**

Follow the syntax described in "Syntax of crontab File Entries" on page 114. The crontab file will be placed in the /var/spool/cron/crontabs directory.

3 **Verify your crontab file changes.**

```
# crontab -l [username]
```

Example 8–1 Creating a crontab File

The following example shows how to create a crontab file for another user.

```
# crontab -e jones
```

The following command entry added to a new crontab file automatically removes any log files from the user's home directory at 1:00 a.m. every Sunday morning. Because the command entry does not redirect output, redirect characters are added to the command line after *.log. Doing so ensures that the command executes properly.

```
# This command helps clean up user accounts.
1 0 * * 0 rm /home/jones/*.log > /dev/null 2>&1
```

▼ How to Verify That a crontab File Exists

- **To verify that a crontab file exists for a user, use the ls -l command in the /var/spool/cron/crontabs directory. For example, the following output shows that crontab files exist for users jones and smith.**

```
$ ls -l /var/spool/cron/crontabs
-rw-r--r--  1 root     sys          190 Feb 26 16:23 adm
-rw-------  1 root     staff        225 Mar  1  9:19 jones
-rw-r--r--  1 root     root        1063 Feb 26 16:23 lp
-rw-r--r--  1 root     sys          441 Feb 26 16:25 root
-rw-------  1 root     staff         60 Mar  1  9:15 smith
-rw-r--r--  1 root     sys          308 Feb 26 16:23 sys
```

Verify the contents of user's crontab file by using the crontab -l command as described in "How to Display a crontab File" on page 117.

Displaying crontab Files

The crontab -l command displays the contents of a crontab file much the same way that the cat command displays the contents of other types of files. You do not have to change the directory to /var/spool/cron/crontabs directory (where crontab files are located) to use this command.

By default, the crontab -l command displays your own crontab file. To display crontab files that belong to other users, you must be superuser.

How to Display a crontab File

Before You Begin Become superuser or assume an equivalent role to display a crontab file that belongs to root or another user. Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

You do not need to become superuser or assume an equivalent role to display your own crontab file.

- **Display the crontab file.**

```
$ crontab -l [username]
```

where *username* specifies the name of the user's account for which you want to display a crontab file. Displaying another user's crontab file requires superuser privileges.

Caution – If you accidentally type the crontab command with no option, press the interrupt character for your editor. This character allows you to quit without saving changes. If you instead saved changes and exited the file, the existing crontab file would be overwritten with an empty file.

Example 8–2

Displaying a crontab File

This example shows how to use the crontab -l command to display the contents of the user's default crontab file.

```
$ crontab -l
13 13 * * * chmod g+w /home1/documents/*.book > /dev/null 2>&1
```

Example 8–3

Displaying the Default root crontab file.

This example shows how to display the default root crontab file.

```
$ suPassword:
Sun Microsystems Inc.   SunOS 5.10      s10_51  May 2004
# crontab -l
#ident  "@(#)root       1.19    98/07/06 SMI"   /* SVr4.0 1.1.3.1       */
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
```

Example 8–4 Displaying the crontab File of Another User

This example shows how to display the crontab file that belongs to another user.

```
$ su
Password:
Sun Microsystems Inc.   SunOS 5.10      s10_51  May 2004
# crontab -l jones
13 13 * * * cp /home/jones/work_files /usr/backup/. > /dev/null 2>&1
```

Removing crontab Files

By default, crontab file protections are set up so that you cannot inadvertently delete a crontab file by using the rm command. Instead, use the crontab -r command to remove crontab files.

By default, the crontab -r command removes your own crontab file.

You do not have to change the directory to /var/spool/cron/crontabs (where crontab files are located) to use this command.

▼ How to Remove a crontab File

Before You Begin Become superuser or assume an equivalent role to remove a crontab file that belongs to root or another user. Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

You do not need to become superuser or assume an equivalent role to remove your own crontab file.

1. **Remove the crontab file.**

```
$ crontab -r [username]
```

where *username* specifies the name of the user's account for which you want to remove a crontab file. Removing crontab files for another user requires superuser privileges.

Caution – If you accidentally type the crontab command with no option, press the interrupt character for your editor. This character allows you to quit without saving changes. If you instead saved changes and exited the file, the existing crontab file would be overwritten with an empty file.

2. **Verify that the crontab file has been removed.**

```
# ls /var/spool/cron/crontabs
```

Example 8–5 Removing a crontab File

The following example shows how user smith uses the crontab -r command to remove his crontab file.

```
$ ls /var/spool/cron/crontabs
adm     jones     lp     root    smith    sys     uucp
$ crontab -r
$ ls /var/spool/cron/crontabs
adm     jones     lp     root    sys    uucp
```

Controlling Access to the crontab Command

You can control access to the crontab command by using two files in the /etc/cron.d directory: cron.deny and cron.allow. These files permit only specified users to perform crontab command tasks such as creating, editing, displaying, or removing their own crontab files.

The cron.deny and cron.allow files consist of a list of user names, one user name per line.

These access control files work together as follows:

- If cron.allow exists, only the users who are listed in this file can create, edit, display, or remove crontab files.
- If cron.allow does not exist, all users can submit crontab files, except for users who are listed in cron.deny.
- If neither cron.allow nor cron.deny exists, superuser privileges are required to run the crontab command.

Superuser privileges are required to edit or create the cron.deny and cron.allow files.

The cron.deny file, which is created during SunOS software installation, contains the following user names:

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

None of the user names in the default cron.deny file can access the crontab command. You can edit this file to add other user names that will be denied access to the crontab command.

No default cron.allow file is supplied. So, after Solaris software installation, all users (except users who are listed in the default cron.deny file) can access the crontab command. If you create a cron.allow file, only these users can access the crontab command.

▼ How to Deny crontab Command Access

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Edit the /etc/cron.d/cron.deny file and add user names, one user per line. Include users who will be denied access to the crontab commands.**

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

3 **Verify that the /etc/cron.d/cron.deny file contains the new entries.**

```
# cat /etc/cron.d/cron.deny
daemon
bin
nuucp
listen
nobody
noaccess
```

▼ How to Limit crontab Command Access to Specified Users

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Create the /etc/cron.d/cron.allow file.**

3 **Add the root user name into the cron.allow file.**

If you do not add root to the file, superuser access to crontab commands will be denied.

4 **Add the user names, one user name per line. Include users that will be allowed to use the crontab command.**

```
root
username1
username2
username3
.
.
.
```

Example 8–6 Limiting crontab Command Access to Specified Users

The following example shows a cron.deny file that prevents user names jones, temp, and visitor from accessing the crontab command.

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
temp
visitor
```

The following example shows a cron.allow file. The users root, jones, lp, and smith are the only users who can access the crontab command.

```
$ cat /etc/cron.d/cron.allow
root
jones
lp
smith
```

How to Verify Limited crontab Command Access

To verify if a specific user can access the crontab command, use the crontab -l command while you are logged into the user account.

```
$ crontab -l
```

If the user can access the crontab command, and already has created a crontab file, the file is displayed. Otherwise, if the user can access the crontab command but no crontab file exists, a message similar to the following message is displayed:

```
crontab: can't open your crontab file
```

Either this user either is listed in the cron.allow file (if the file exists), or the user is not listed in the cron.deny file.

If the user cannot access the crontab command, the following message is displayed whether or not a previous crontab file exists:

```
crontab: you are not authorized to use cron. Sorry.
```

This message means that either the user is not listed in the cron.allow file (if the file exists), or the user is listed in the cron.deny file.

Using the at Command (Task Map)

Task	Description	For Instructions
Create an at job.	Use the atcommand to do the following: ■ Start the at utility from the command line. ■ Type the commands or scripts that you want to execute, one per line. ■ Exit the at utility and save the job.	"How to Create an at Job" on page 124
Display the at queue.	User the atq command to display the at queue.	"How to Display the at Queue" on page 125
Verify an at job.	Use the atq command to confirm that at jobs that belong to a specific user have been submitted to the queue.	"How to Verify an at Job" on page 125
Display at jobs.	Use the at -l[*job-id* to display at jobs. that have been submitted to the queue.	"How to Display at Jobs" on page 126

Task	Description	For Instructions
Remove at jobs.	Use the at -r [*job-id*] command to remove at jobs from the queue.	"How to Remove at Jobs" on page 126
Deny access to the at command.	To deny users access to the atcommand, edit the /etc/cron.d/at.deny file.	"How to Deny Access to the at Command" on page 127

Scheduling a Single System Task (at)

The following sections describe how to use the at command to perform the following tasks:

- Schedule jobs (command and scripts) for execution at a later time
- How to display and remove these jobs
- How to control access to the at command

By default, users can create, display, and remove their own at job files. To access at files that belong to root or other users, you must have superuser privileges.

When you submit an at job, it is assigned a job identification number along with the .a extension. This designation becomes the job's file name, as well as its queue number.

Description of the at Command

Submitting an at job file involves these steps:

1. Invoking the at utility and specifying a command execution time.
2. Typing a command or script to execute later.

Note – If output from this command or script is important, be sure to direct the output to a file for later examination.

For example, the following at job removes core files from the user account smith near midnight on the last day of July.

```
$ at 11:45pm July 31
at> rm /home/smith/*core*
at> Press Control-d
commands will be executed using /bin/csh
job 933486300.a at Tue Jul 31 23:45:00 2004
```

Controlling Access to the `at` Command

You can set up a file to control access to the `at` command, permitting only specified users to create, remove, or display queue information about their `at` jobs. The file that controls access to the `at` command, `/etc/cron.d/at.deny`, consists of a list of user names, one user name per line. The users who are listed in this file cannot access `at` commands.

The `at.deny` file, which is created during SunOS software installation, contains the following user names:

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

With superuser privileges, you can edit the `at.deny` file to add other user names whose `at` command access you want to restrict.

▼ How to Create an `at` Job

1. **Start the `at` utility, specifying the time you want your job executed.**

   ```
   $ at [-m] time [date]
   ```

   `-m` Sends you email after the job is completed.

   time Specifies the hour that you want to schedule the job. Add `am` or `pm` if you do not specify the hours according to the 24-hour clock. Acceptable keywords are `midnight`, `noon`, and `now`. Minutes are optional.

   date Specifies the first three or more letters of a month, a day of the week, or the keywords `today` or `tomorrow`.

2. **At the `at` prompt, type the commands or scripts that you want to execute, one per line.**

   You may type more than one command by pressing Return at the end of each line.

3. **Exit the `at` utility and save the `at` job by pressing Control-D.**

   Your `at` job is assigned a queue number, which is also the job's file name. This number is displayed when you exit the `at` utility.

Example 8–7 Creating an at Job

The following example shows the at job that user jones created to remove her backup files at 7:30 p.m. She used the -m option so that she would receive an email message after her job completed.

```
$ at -m 1930
at> rm /home/jones/*.backup
at> Press Control-D
job 897355800.a at Thu Jul  12 19:30:00 2004
```

She received a email message which confirmed the execution of her at job.

```
Your “at” job “rm /home/jones/*.backup”
completed.
```

The following example shows how jones scheduled a large at job for 4:00 a.m. Saturday morning. The job output was directed to a file named big.file.

```
$ at 4 am Saturday
at> sort -r /usr/dict/words > /export/home/jones/big.file
```

▼ How to Display the at Queue

- **To check your jobs that are waiting in the at queue, use the atq command. This command displays status information about the at jobs that you have created.**

  ```
  $ atq
  ```

▼ How to Verify an at Job

- **To verify that you have created an at job, use the atq command. In the following example, the atq command confirms that at jobs that belong to jones have been submitted to the queue.**

  ```
  $ atq
  Rank      Execution Date     Owner     Job         Queue   Job Name
    1st   Jul 12, 2004 19:30   jones  897355800.a     a     stdin
    2nd   Jul 14, 2004 23:45   jones  897543900.a     a     stdin
    3rd   Jul 17, 2004 04:00   jones  897732000.a     a     stdin
  ```

How to Display at Jobs

- **To display information about the execution times of your at jobs, use the at -l command.**

```
$ at -l [job-id]
```

where the -l *job-id* option identifies the identification number of the job whose status you want to display.

Example 8–8 Displaying at Jobs

The following example shows output from the at -l command, which provides information on the status of all jobs submitted by a user.

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2004
897355800.a    Thu Jul 12 19:30:00 2004
897732000.a    Tue Jul 17 04:00:00 2004
```

The following example shows the output that is displayed when a single job is specified with the at -l command.

```
$ at -l 897732000.a
897732000.a        Tue Jul 17 04:00:00 2004
```

How to Remove at Jobs

Before You Begin Become superuser or assume an equivalent role to remove an at job that belongs to root or another user. Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

You do not need to become superuser or assume an equivalent role to remove your own at job.

1. **Remove the at job from the queue before the job is executed.**

```
$ at -r [job-id]
```

where the -r *job-id* option specifies the identification number of the job you want to remove.

2. **Verify that the at job is removed by using the at -l (or the atq) command.**

The at -l command displays the jobs remaining in the at queue. The job whose identification number you specified should not appear.

```
$ at -l [job-id]
```

Example 8–9 Removing at Jobs

In the following example, a user wants to remove an at job that was scheduled to execute at 4 a.m. on July 17th. First, the user displays the at queue to locate the job identification number. Next, the user removes this job from the at queue. Finally, the user verifies that this job has been removed from the queue.

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2003
897355800.a    Thu Jul 12 19:30:00 2003
897732000.a    Tue Jul 17 04:00:00 2003
$ at -r 897732000.a
$ at -l 897732000.a
at: 858142000.a: No such file or directory
```

▼ How to Deny Access to the at Command

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Edit the /etc/cron.d/at.deny file and add the names of users, one user name per line, that will be prevented from using the at commands.**

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

Example 8–10 Denying at Access

The following example shows an at.deny file that has been edited so that the users smith and jones cannot access the at command.

```
$ cat at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
smith
```

How to Verify That at Command Access Is Denied

- **To verify that a username was added correctly to the /etc/cron.d/at.deny file, use the at -l command while logged in as the user. If the user smith cannot access the at command, the following message is displayed.**

```
# su smith
Password:
$ at -l
at: you are not authorized to use at.  Sorry.
```

Likewise, if the user tries to submit an at job, the following message is displayed:

```
$ at 2:30pm
at: you are not authorized to use at.  Sorry.
```

This message confirms that the user is listed in the at.deny file.

If at command access is allowed, then the at -l command returns nothing.

◆ ◆ ◆ CHAPTER 9

Managing System Accounting (Tasks)

This chapter describes how to set up and maintain system accounting.

This is a list of the overview information in this chapter.

- "What is System Accounting?" on page 130
- "Setting Up System Accounting" on page 135

For information on using extended accounting, see Chapter 4, "Extended Accounting (Overview)," in *System Administration Guide: Solaris Containers-Resource Management and Solaris Zones*.

For information on the step-by-step procedures that are associated with system accounting, see "System Accounting (Task Map)" on page 134.

For reference information about the various system accounting reports, see Chapter 10, "System Accounting (Reference)."

What's New in System Accounting

This section describes new or changed features in system accounting in the Solaris release. For a complete listing of new Solaris features and a description of Solaris releases, see *Solaris 10 What's New*.

Solaris Process Accounting and Statistics Improvements

Solaris 10: Changes have been made to the internals of the load averaging, `cpu` `usr/sys/idle`, and accounting functions. Microstate accounting has replaced the old accounting mechanism and is enabled by default all of the time. As a result, you might notice slightly different process usage and timing statistics.

Switching to microstate accounting provides substantially more accurate data about user processes and the amount of time they spend in various states. In addition, this information is used to generate more accurate load averages and statistics from the /proc file system. For more information, see the proc(4) man page.

What is System Accounting?

System accounting software in the Solaris OS is a set of programs that enables you to collect and record data about user connect time, CPU time charged to processes, and disk usage. Once you collect this data, you can generate reports and charge fees for system usage.

You can use system accounting on a daily or monthly basis. Or, you can tack disk usage per user.

You can use the accounting programs to perform these tasks:

- Monitor system usage
- Locate and correct performance problems
- Maintain system security

After you set up the system accounting programs, they run mostly on their own.

How System Accounting Works

Automatic accounting is set up by first putting the accounting startup script into root's crontab file. The accounting startup script can then be started automatically by the cron command.

The following overview describes the system accounting process.

1. Between system startup and shutdown, raw data about system use (such as user logins, running processes, and data storage) are collected in accounting files.
2. Periodically (usually once a day), the /usr/lib/acct/runacct script processes the various accounting files and produces both cumulative summary files and daily accounting reports. Then, the /usr/lib/acct/prdaily script prints the daily reports.

   For more information about the runacct script, see "runacct Script" on page 143.
3. Monthly, you can process and print the cumulative runacct summary files by executing the monacct script. The summary reports produced by the monacct script provide an efficient means for billing users on a monthly or other fiscal basis.

System Accounting Components

The system accounting software provides C language programs and shell scripts that organize data into summary files and reports. These programs reside in the /usr/lib/acct directory. The accounting reports reside in the /var/adm/acct directory.

Daily accounting can help you perform four types of auditing:

- Connect accounting
- Process accounting
- Disk accounting
- Fee calculations

Connect Accounting

Connect accounting enables you to determine the following information:

- The length of time a user was logged in
- How the `tty` lines are being used
- The number of reboots on your system
- How many times the accounting software was turned off and on

To provide this information on connect sessions, the system stores the following data

- Record of time adjustments
- Boot times
- Number of times the accounting software was turned off and on
- Changes in run levels
- The creation of user processes (`login` processes and `init` processes)
- The terminations of processes

These records are produced from the output of system programs such as `date`, `init`, `login`, `ttymon`, and `acctwtmp`. They are stored in the `/var/adm/wtmpx` file.

Entries in the `wtmpx` file can contain the following information:

- Login name
- Device name
- Process ID
- Entry type
- Time stamp that denotes when the entry was made

Process Accounting

Process accounting enables you to keep track of the following data about each process that runs on your system:

- User IDs and group IDs of users using the process
- Beginning times and elapsed times of the process
- CPU time for the process (user time and system time)
- Amount of memory used by the process
- Commands run by the process
- The `tty` that controls the process

Every time a process terminates, the exit program collects this information and writes it to the /var/adm/pacct file.

Disk Accounting

Disk accounting enables you to gather and format the following data about the files each user has on disks:

- User name and user ID of the user
- Number of blocks that are used by the user's files

This data is collected by the /usr/lib/acct/dodisk shell script at intervals that are determined by the entry you add to the /var/spool/cron/crontabs/root file. In turn, the dodisk script invokes the acctdisk and acctdusg commands. These commands gather disk usage by login name.

Caution – Information gathered by running the dodisk script is stored in the /var/adm/acct/nite/disktacct file. This information is overwritten the next time the dodisk script is run. Therefore, avoid running the dodisk script twice in the same day.

The acctdusg command might overcharge for files that are written randomly, which can create holes in the files. This problem occurs because the acctdusg command does not read the indirect blocks of a file when determining the file size. Rather, the acctdusg command determines the file size by checking the current file size value in the file's inode.

Fee Calculations

The chargefee utility stores charges for special services that are provided to a user in the /var/adm/fee file. A special service, for example, is file restoration. Each entry in the file consists of a user login name, user ID, and the fee. This file is checked by the runacct script every day, and new entries are merged into the accounting records. For instructions on running the chargefee script to bill users, see "How to Bill Users" on page 138.

How Daily Accounting Works

Here is a step-by-step summary of how daily accounting works:

1. When the system is switched into multiuser mode, the /usr/lib/acct/startup program is executed. The startup program executes several other programs that invoke daily accounting.
2. The acctwtmp program adds a "boot" record to the /var/adm/wtmpx file. In this record, the system name is shown as the user name in the wtmpx record. The following table summarizes how the raw accounting data is gathered and where it is stored.

File in `/var/adm`	Information Stored	Written By	Format
`wtmpx`	Connect sessions	`login`, `init`	Binary
	Changes	`date`	Binary
	Reboots	`acctwtmp`	Binary
	Shutdowns	`shutacct`	Binary
`pacctn`	Processes	Kernel (when the process ends)	Binary
		`turnacct switch` (which creates a new file when the old file reaches 500 blocks)	Binary
`fee`	Special charges	`chargefee`	ASCII
`acct/nite/disktacct`	Disk space used	`dodisk`	Binary

3. The `turnacct` script, invoked with the `-on` option, begins process accounting. Specifically, the `turnacct` script executes the `accton` program with the `/var/adm/pacct` argument.
4. The remove shell script "cleans up" the saved `pacct` and `wtmpx` files that are left in the `sum` directory by the `runacct` script.
5. The `login` and `init` programs record connect sessions by writing records into the `/var/adm/wtmpx` file. Date changes (using date with an argument) are also written to the `/var/adm/wtmpx` file. Reboots and shutdowns using the `acctwtmp` command are also recorded in the `/var/adm/wtmpx` file.
6. When a process ends, the kernel writes one record per process, using the `acct.h` format, in the `/var/adm/pacct` file.

   Every hour, the `cron` command executes the `ckpacct` script to check the size of the `/var/adm/pacct` file. If the file grows beyond 500 blocks (default), the `turnacct switch` command is executed. (The program moves the `pacct` file to the `pacct`*n* file and creates a new file.) The advantage of having several smaller `pacct` files becomes apparent when you try to restart the `runacct` script if a failure occurs when processing these records.
7. The `runacct` script is executed by the `cron` command each night. The `runacct` script processes the accounting files to produce command summaries and usage summaries by user name. These accounting files are processed: `/var/adm/pacct`*n*, `/var/adm/wtmpx`, `/var/adm/fee`, and `/var/adm/acct/nite/disktacct`.
8. The `/usr/lib/acct/prdaily` script is executed on a daily basis by the `runacct` script to write the daily accounting information in the `/var/adm/acct/sum/rprt`*MMDD* files.

9. The `monacct` script should be executed on a monthly basis (or at intervals you determine, such as at the end of every fiscal period). The `monacct` script creates a report that is based on data stored in the `sum` directory that has been updated daily by the `runacct` script. After creating the report, the `monacct` script "cleans up" the `sum` directory to prepare the directory's files for the new `runacct` data.

What Happens if the System Shuts Down

If the system is shut down by using the `shutdown` command, the `shutacct` script is executed automatically. The `shutacct` script writes a *reason record* into the `/var/adm/wtmpx` file and turns off process accounting.

System Accounting (Task Map)

Task	Description	For Instructions
Set up system accounting.	Set up system accounting by performing the following tasks: ■ Create the `/etc/rc0.d/K22acct` and `/etc/rc2.d/S22acct` files. ■ Modify the `/var/spool/cron/crontabs/adm` and `/var/spool/cron/crontabs/root` crontab files.	"How to Set Up System Accounting" on page 136
Bill users.	Run the `/usr/lib/acct/chargefee` *username amount* command.	"How to Bill Users" on page 138
Fix a corrupted `wtmpx` file.	Convert the `wtmpx` file from binary to ASCII format.	"How to Fix a Corrupted `wtmpx` File" on page 139
Fix `tacct` errors.	Run the `prtacct` script to check the `/var/adm/acct/sum/tacctprev` file. Then, patch the latest`/var/adm/acct/sum/tacct`*MMDD* file. You will need to re-create the `/var/adm/acct/sum/tacct` file.	"How to Fix `tacct` Errors" on page 139
Restart the `runacct` script.	Remove the `lastdate` file and any lock files. Then, manually restart the `runacct` script.	"How to Restart the `runacct` Script" on page 141
Disable system accounting temporarily.	Edit theadm `crontab` file to stop the `ckpacct`, `runacct`, and `monacct` programs from running.	"How to Temporarily Stop System Accounting" on page 141

Task	Description	For Instructions
Disable system accounting permanently.	Delete the entries for the ckpacct, runacct, and monacct programs in the adm and crontab files.	"How to Permanently Disable System Accounting" on page 142

Setting Up System Accounting

You can set up system accounting to run while the system is in multiuser mode (Run Level 2). Generally, this task involves these steps:

1. Creating the /etc/rc0.d/K22acct and /etc/rc2.d/S22acct startup scripts
2. Modifying the /var/spool/cron/crontabs/adm and /var/spool/cron/crontabs/root crontab files

The following table describes the default accounting scripts.

TABLE 9–1 Default Accounting Scripts

Purpose	Accounting Script	Man Page	Run Frequency
Checks the size of the /usr/adm/pacct log file and makes sure that it does not get too large.	ckpacct	acctsh(1M)	Periodically
Processes connect, disk, and fee accounting information. You can remove from this script the commands for the accounting features you do not want processed.	runacct	runacct(1M)	Daily
Generates fiscal accounting summary reports on a monthly basis. You can determine how often this script is run. You can remove from this script the commands for the accounting features you do not want to use.	monacct	acctsh(1M)	On a fiscal basis

You can choose which accounting scripts run by default. After these entries have been added to the crontab files, system accounting should run automatically.

▼ How to Set Up System Accounting

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **If necessary, install the SUNWaccr and SUNWaccu packages on your system by using the pkgadd command.**

3. **Install /etc/init.d/acct as the startup script for Run Level 2.**

   ```
   # ln /etc/init.d/acct /etc/rc2.d/S22acct
   ```

4. **Install /etc/init.d/acct as the stop script for Run Level 0.**

   ```
   # ln /etc/init.d/acct /etc/rc0.d/K22acct
   ```

5. **Add the following lines to the adm crontab file to start the ckpacct, runacct, and monacct scripts automatically.**

   ```
   # EDITOR=vi; export EDITOR
   # crontab -e adm
   0 * * * * /usr/lib/acct/ckpacct
   30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
   30 7 1 * * /usr/lib/acct/monacct
   ```

6. **Add the following line to the root crontab file to start the dodisk script automatically.**

   ```
   # crontab -e
   30 22 * * 4 /usr/lib/acct/dodisk
   ```

7. **Edit /etc/acct/holidays to include national holidays and local holidays.**

   For more information, see the holidays(4) man page and the example that follows.

8. **Reboot the system, or start system accounting manually by typing:**

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

Example 9–1 Setting Up Accounting (adm crontab)

This modified adm crontab contains entries for the ckpacct, runacct, and monacct scripts.

```
#ident  "@(#)adm        1.5     92/07/14 SMI"    /* SVr4.0 1.2   */
#
# The adm crontab file should contain startup of performance
# collection if the profiling and performance feature has been
# installed.
0 * * * * /usr/lib/acct/ckpacct
```

```
30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
30 7 1 * * /usr/lib/acct/monacct
```

Example 9–2 Setting Up Accounting (root crontab)

This modified root crontab contains entries for the dodisk program.

```
#ident  "@(#)root       1.19    98/07/06 SMI"   /* SVr4.0 1.1.3.1       */
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
30 22 * * 4 /usr/lib/acct/dodisk
```

Example 9–3 Setting Up Accounting (/etc/acct/holidays)

The following example shows a sample /etc/acct/holidays file.

```
* @(#)holidays    January 1, 2004
*
* Prime/Nonprime Table for UNIX Accounting System
*
* Curr    Prime    Non-Prime
* Year    Start    Start
*
  2004    0800    1800
*
* only the first column (month/day) is significant.
*
* month/day    Company
*         Holiday
*
1/1        New Years Day
7/4        Indep. Day
12/25        Christmas
```

Billing Users

If you provide special user services by request, you might want to bill users by running the chargefee utility. Special services include restoring files or remote printing. The chargefee utility records charges in the /var/adm/fee file. Each time the runacct utility is executed, new entries are merged into the total accounting records.

See the acctsh(1M) man page for more information.

▼ How to Bill Users

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Charge a user for special services.**

```
# /usr/lib/acct/chargefee username amount
```

username Is the user account you want to bill.

amount Specifies the number of units to bill the user. This value is an arbitrary unit that you set to charge users based on some task such as printing or restoring a file. You would have to write a script that invokes the chargefee utility and charges a user for a specific task.

Example 9–4 Billing Users

In the following example, the user print_customer is charged 10 units.

```
# /usr/lib/acct/chargefee print_customer 10
```

Maintaining Accounting Information

This section describes how to fix corrupted system accounting files and how to restart the runacct script.

Fixing Corrupted Files and wtmpx Errors

Unfortunately, system accounting is not foolproof. Occasionally, a file becomes corrupted or lost. Some files can simply be ignored or restored from backup. However, certain files must be fixed to maintain the integrity of system accounting.

The wtmpx files seem to cause the most problems in the daily operation of system accounting. When the date is changed manually and the system is in multiuser mode, a set of date change records is written to the /var/adm/wtmpx file. The wtmpfix utility is designed to adjust the time stamps in the wtmp records when a date change is encountered. However, some combinations of date changes and reboots slip through the wtmpfix utility and cause the acctcon program to fail.

▼ How to Fix a Corrupted wtmpx File

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Change to the /var/adm directory.**

3 **Convert the wtmpx file from binary format to ASCII format.**

```
# /usr/lib/acct/fwtmp < wtmpx > wtmpx.ascii
```

4 **Edit the wtmpx.ascii file to delete the corrupted records.**

5 **Convert the wtmpx.ascii file back to a binary file.**

```
# /usr/lib/acct/fwtmp -ic < wtmpx.ascii > wtmpx
```

See the fwtmp(1M) man page for more information.

Fixing tacct Errors

The integrity of the /var/adm/acct/sum/tacct file is important if you are charging users for system resources. Occasionally, unusual tacct records appear with negative numbers, duplicate user IDs, or a user ID of 65535. First, check the /var/adm/acct/sum/tacctprev file by using the prtacct script to print the file. If the contents look all right, patch the latest /var/adm/acct/sum/tacct*MMDD* file. Then, re-create the /var/adm/acct/sum/tacct file. The following steps outline a simple patch procedure.

▼ How to Fix tacct Errors

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Change to the** `/var/adm/acct/sum` **directory.**

3 **Convert the** `tacct`*MMDD* **file from binary format to ASCII format.**

```
# /usr/lib/acct/acctmerg -v < tacctMMDD > xtacct
```

MMDD is pair of two-digit numbers that represent the month and day.

4 **Edit the** `xtacct` **file, removing corrupted records and writing duplicate records to another file.**

5 **Convert the** `xtacct` **file from ASCII format to binary format.**

```
# /usr/lib/acct/acctmerg -i < xtacct > tacctMMDD
```

6 **Merge the files** `tacctprev` **and** `tacct.`*MMDD* **into the** `tacct` **file.**

```
# /usr/lib/acct/acctmerg < tacctprev tacctMMDD > tacct
```

Restarting the `runacct` Script

The `runacct` script can fail for several reasons.

The following are the most common reasons:

- A system crash
- The `/var` directory is running out of space
- A corrupted `wtmpx` file

If the `active.`*MMDD* file exists, check it first for error messages. If the `active` and `lock` files exist, check the `fd2log` file for any relevant messages.

Run without arguments, the `runacct` script assumes that this invocation is the first invocation of the day. The argument *MMDD* is necessary if the `runacct` script is being restarted and specifies the month and day for which the `runacct` script reruns the accounting. The entry point for processing is based on the contents of the `statefile` file. To override the `statefile` file, include the desired state on the command line. For a description of the available states, see the `runacct`(1M) man page.

Caution – When you run the `runacct` program manually, be sure to run it as user `adm`.

▼ How to Restart the runacct Script

1 **Change directories to the /var/adm/acct/nite directory.**

```
$ cd /var/adm/acct/nite
```

2 **Remove the lastdate file and any lock* files, if any.**

```
$ rm lastdate lock*
```

The `lastdate` file contains the date that the `runacct` program was last run. Restarting the `runacct` script in the next step re-creates this file.

3 **Restart the runacct script.**

```
$ /usr/lib/acct/runacct MMDD [state] 2> /var/adm/acct/nite/fd2log &
```

MMDD Is the month and day specified by two-digit numbers.

state Specifies a state, or starting point, where the `runacct` script processing should begin.

Stopping and Disabling System Accounting

You can temporarily stop system accounting or permanently disable it.

▼ How to Temporarily Stop System Accounting

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Edit the adm crontab file to stop the ckpacct, runacct, and monacct programs from running by commenting out the appropriate lines.**

```
# EDITOR=vi; export EDITOR
# crontab -e adm
#0 * * * * /usr/lib/acct/ckpacct
#30 2 * * * /usr/lib/acct/runacct 2> /var/adm/acct/nite/fd2log
#30 7 1 * * /usr/lib/acct/monacct
```

3 **Edit the root crontab file to stop the dodisk program from running by commenting out the appropriate line.**

```
# crontab -e
#30 22 * * 4 /usr/lib/acct/dodisk
```

4. **Stop the system accounting program.**

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

5. **(Optional) Remove the newly added comment symbols from the `crontab` files.**

6. **Restart the system accounting program to re-enable system accounting.**

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

▼ How to Permanently Disable System Accounting

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Edit the `adm` `crontab` file and delete the entries for the `ckpacct`, `runacct`, and `monacct` programs.**

```
# EDITOR=vi; export EDITOR
# crontab -e adm
```

3. **Edit the `root` `crontab` file and delete the entries for the `dodisk` program.**

```
# crontab -e
```

4. **Remove the startup script for Run Level 2.**

```
# unlink /etc/rc2.d/S22acct
```

5. **Remove the stop script for Run Level 0.**

```
# unlink /etc/rc0.d/K22acct
```

6. **Stop the system accounting program.**

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

CHAPTER 10

System Accounting (Reference)

This chapter provides reference information about system accounting.

This is a list of the reference information in this chapter.

For more information about system accounting tasks, see Chapter 9, "Managing System Accounting (Tasks)."

runacct Script

The main daily accounting script, runacct, is normally invoked by the cron command outside of normal business hours. The runacct script processes connect, fee, disk, and process accounting files. This script also prepares daily and cumulative summary files for use by the prdaily and monacct scripts for billing purposes.

The runacct script takes care not to damage files if errors occur.

A series of protection mechanisms that are used to perform the following tasks:

- Recognize an error
- Provide intelligent diagnostics
- Complete processing in such a way that the runacct script can be restarted with minimal intervention

This script records its progress by writing descriptive messages to the active file. Files used by the runacct script are assumed to be in the /var/adm/acct/nite directory, unless otherwise noted. All diagnostic output during the execution of the runacct script is written to the fd2log file.

When the runacct script is invoked, it creates the lock and lock1 files. These files are used to prevent simultaneous execution of the runacct script. The runacct program prints an error message if these files exist when it is invoked. The lastdate file contains the month and day the runacct script was last invoked, and is used to prevent more than one execution per day.

If the runacct script detects an error, the following occurs:

- A message is written to the console
- Email is sent to root and adm
- Locks might be removed
- Diagnostics are saved
- Execution is ended

For instructions on how to restart the runacct script, see "How to Restart the runacct Script" on page 141.

To allow the runacct script to be restarted, processing is broken down into separate re-entrant states. The statefile file is used to track the last state completed. When each state is completed, the statefile file is updated to reflect the next state. After processing for the state is complete, the statefile file is read and the next state is processed. When the runacct script reaches the CLEANUP state, it removes the locks and ends. States are executed as shown in the following table.

TABLE 10–1 States of the runacct Script

State	Description
SETUP	The turnacct switch command is executed to create a new pacct file. The /var/adm/pacct*n* process accounting files (except for the pacct file) are moved to the /var/adm/Spacct*n*.*MMDD* files. The /var/adm/wtmpx file is moved to the /var/adm/acct/nite/wtmp.*MMDD* file (with the current time record added on the end) and a new /var/adm/wtmp file is created. The closewtmp and utmp2wtmp programs add records to the wtmp.*MMDD* file and the new wtmpx file to account for users who are currently logged in.
WTMPFIX	The wtmpfix program checks the wtmp.*MMDD* file in the nite directory for accuracy. Because some date changes cause the acctcon program to fail, the wtmpfix program attempts to adjust the time stamps in the wtmpx file if a record of a date change appears. This program also deletes any corrupted entries from the wtmpx file. The fixed version of the wtmp.*MMDD* file is written to the tmpwtmp file.
CONNECT	The acctcon program is used to record connect accounting records in the file ctacct.*MMDD*. These records are in tacct.h format. In addition, the acctcon program creates the lineuse and reboots files. The reboots file records all the boot records found in the wtmpx file.

TABLE 10–1 States of the runacct Script *(Continued)*

State	Description
PROCESS	The acctprc program is used to convert the /var/adm/Spacct*n.MMDD* process accounting files into complete accounting records in the ptacct*n.MMDD* files. The Spacct and ptacct files are correlated by number so that if the runacct script fails, the Spacct files are not processed.
MERGE	The acctmerg program merges the process accounting records with the connect accounting records to form the daytacct file.
FEES	The acctmerg program merges ASCII tacct records from the fee file into the daytacct file.
DISK	The dodisk script produces the disktacct file. If the dodisk script has been run, which produces the disktacct file, the DISK program merges the file into the daytacct file and moves the disktacct file to the /tmp/disktacct.*MMDD* file.
MERGETACCT	The acctmerg program merges the daytacct file with the sum/tacct file, the cumulative total accounting file. Each day, the daytacct file is saved in the sum/tacct.*MMDD* file so that the sum/tacct file can be re-created if it is corrupted or lost.
CMS	The acctcms program is run several times. This program is first run to generate the command summary by using the Spacct*n* files and write the data to the sum/daycms file. The acctcms program is then run to merge the sum/daycms file with the sum/cms cumulative command summary file. Finally, the acctcms program is run to produce nite/daycms and nite/cms, the ASCII command summary files from the sum/daycms and sum/cms files, respectively. The lastlogin program is used to create the /var/adm/acct/sum/loginlog log file. This file reports when each user last logged in. If the runacct script is run after midnight, the dates showing the time last logged in by some users will be incorrect by one day.
USEREXIT	Any installation-dependent (local) accounting program can be run at this point. The runacct script expects this program to be called the /usr/lib/acct/runacct.local program.
CLEANUP	This state cleans up temporary files, runs the prdaily script and saves its output in the sum/rpt.*MMDD* file, removes the locks, and then exits.

Caution – When restarting the runacct script in the CLEANUP state, remove the last ptacct file because this file will not be complete.

Daily Accounting Reports

The runacct shell script generates five basic reports upon each invocation. The following table describes these reports.

TABLE 10–2 Daily Accounting Reports

Report Type	Description
"Daily Report" on page 146	Shows terminal line utilization by tty number.
"Daily Usage Report" on page 147	Indicates usage of system resources by users (listed in order of user ID).
"Daily Command Summary" on page 148	Indicates usage of system resources by commands, listed in descending order of memory use. In other words, the command that used the most memory is listed first. This same information is reported for the month in the monthly command summary.
"Monthly Command Summary" on page 150	A cumulative summary that reflects the data accumulated since the last invocation of the monacct program.
"Last Login Report" on page 150	Shows the last time each user logged in (listed in chronological order).

Daily Report

This report gives information about each terminal line used. The following is a sample Daily Report.

```
Jan 16 02:30 2004  DAILY REPORT FOR venus Page 1


from Mon Jan 15 02:30:02 2004
to   Tue Oan 16 02:30:01 2004
1       runacct
1       acctcon

TOTAL DURATION IS 1440 MINUTES
LINE         MINUTES  PERCENT  # SESS   # ON   # OFF
console      868      60       1        1      2
TOTALS       868      --       1        1      2
```

The from and to lines specify the time period reflected in the report. This time period covers the time the last Daily Report was generated to the time the current Daily Report was generated. Then, the report presents a log of system reboots, shutdowns, power failure recoveries, and any other record written to the /var/adm/wtmpx file by the acctwtmp program. For more information, see the acct(1M) man page.

The second part of the report is a breakdown of terminal line utilization. The TOTAL DURATION tells how long the system was in multiuser mode (accessible through the terminal lines). The following table describes the data provided by the Daily Report.

TABLE 10–3 Daily Report Data

Column	Description
LINE	The terminal line or access port.
MINUTES	The number of minutes that the line was in use during the accounting period.
PERCENT	The TOTAL DURATION divided by the number of MINUTES.
# SESS	The number of times this line or port was accessed for a login session.
# ON	Same as SESS. (This column no longer has meaning. Previously, this column listed the number of times that a line or port was used to log in a user.)
# OFF T	The number of times a user logs out and any interrupts that occur on that line. Generally, interrupts occur on a port when ttymon is first invoked after the system is brought to multiuser mode. If the # OFF exceeds the # SESS by a large factor, the multiplexer, modem, or cable is probably going bad. Or, a bad connection exists somewhere. The most common cause is an unconnected cable dangling from the multiplexer.

During real time, you should monitor the /var/adm/wtmpx file because it is the file from which the connect accounting is derived. If the wtmpx file grows rapidly, execute the following command to see which tty line is the noisiest.

```
# /usr/lib/acct/acctcon -l file < /var/adm/wtmpx
```

If interruption is occurring frequently, general system performance will be affected. Additionally, the wtmp file might become corrupted. To correct this problem, see “How to Fix a Corrupted wtmpx File” on page 139.

Daily Usage Report

The Daily Usage Report breaks down system resource utilization by user. A sample of this report follows.

```
Jan 16 02:30 2004  DAILY USAGE REPORT FOR skisun Page 1


     LOGIN  CPU  (MINS)  KCORE-   MINS    CONNECT  (MINS) DISK   # OF   # OF  # DISK  FEE
UID  NAME   PRIME NPRIME PRIME    NPRIME  PRIME    NPRIME BLOCKS PROCS  SESS  SAMPLES
0    TOTAL  72    148    11006173 51168   26230634 57792  539    330    0     2150    1
0    root   32    76     11006164 33664   26230616 22784  0      0      0     127     0
```

```
4    adm    0     0      22       51      0        0      0      420    0     0       0
101  rimmer 39    72     894385   1766020 539      330    0      1603   1     0       0
```

The following table describes the data provided by the Daily Usage Report.

TABLE 10–4 Daily Usage Report Data

Column	Description
`UID`	User ID number.
`LOGIN NAME`	Login (or user) name of the user. Identifies a user who has multiple login names.
`CPU (MINS)`	Amount of time, in minutes, that the user's process used the central processing unit. Divided into `PRIME` and `NPRIME` (nonprime) utilization. The accounting system's version of this data is located in the `/etc/acct/holidays` file.
`KCORE-MINS`	A cumulative measure of the amount of memory in Kbyte segments per minute that a process uses while running. Divided into `PRIME` and `NPRIME` utilization.
`CONNECT (MINS)`	Amount of time, in minutes, that the a user was logged in to the system, or “real time.” Divided into `PRIME` and `NPRIME` utilization. If these numbers are high while the `# OF PROCS` is low, you can conclude that the user logs in first thing in the morning and hardly touches the terminal the rest of the day.
`DISK BLOCKS`	Output from the `acctdusg` program, which runs the disk accounting programs and merges the accounting records (`daytacct`). For accounting purposes, a block is 512 bytes.
`# OF PROCS`	Number of processes invoked by the user. If large numbers appear, a user might have a shell procedure that has run out of control.
`# OF SESS`	Number of times that a user logged in to the system.
`# DISK SAMPLES`	Number of times that disk accounting was run to obtain the average number of `DISK BLOCKS`.
`FEE`	Often unused field that represents the total accumulation of units charged against the user by the `chargefee` script.

Daily Command Summary

The Daily Command Summary report shows the system resource utilization by command. With this report, you can identify the most heavily used commands. Based on how those commands use system resources, you can then gain insight on how best to tune the system.

These reports are sorted by `TOTAL KCOREMIN`, which is an arbitrary gauge but often useful for calculating drain on a system.

A sample Daily Command Summary follows.

```
                                TOTAL COMMAND SUMMARY
COMMAND    NUMBER       TOTAL    TOTAL      TOTAL    MEAN     MEAN     HOG   CHARS    BLOCKS
NAME         CMDS    KCOREMIN CPU-MIN REAL-MIN  SIZE-K  CPU-MIN  FACTOR TRNSFD   READ

TOTALS       2150  1334999.75  219.59 724258.50 6079.48   0.10   0.00   397338982 419448

netscape       43  2456898.50   92.03  54503.12 26695.51  2.14   0.00   947774912 225568
adeptedi        7    88328.22    4.03    404.12 21914.95  0.58   0.01    93155160   8774
dtmail          1    54919.17    5.33  17716.57 10308.94  5.33   0.00   213843968  40192
acroread        8    31218.02    2.67  17744.57 11682.66  0.33   0.00   331454464  11260
dtwm            1    16252.93    2.53  17716.57 6416.05   2.53   0.00   158662656  12848
dtterm          5     4762.71    1.30  76300.29 3658.93   0.26   0.00    33828352  11604
dtaction       23     1389.72    0.33      0.60 4196.43   0.01   0.55    18653184    539
dtsessio        1     1174.87    0.24  17716.57 4932.97   0.24   0.00    23535616   5421
dtcm            1      866.30    0.18  17716.57 4826.21   0.18   0.00     3012096   6490
```

The following table describes the data provided by the Daily Command Summary.

TABLE 10–5 Daily Command Summary Data

Column	Description
`COMMAND NAME`	Name of the command. All shell procedures are lumped together under the name `sh` because only object modules are reported by the process accounting system. You should monitor the frequency of programs called `a.out` or `core`, or any other unexpected name. You can use the `acctcom` program to determine who executed an oddly named command and if superuser privileges were used.
`NUMBER CMDS`	Total number of times this command was run.
`TOTAL KCOREMIN`	Total cumulative measurement of the Kbyte segments of memory used by a process per minute of run time.
`TOTAL CPU-MIN`	Total processing time this program accumulated.
`TOTAL REAL-MIN`	Total real-time (wall-clock) minutes this program accumulated.
`MEAN SIZE-K`	Mean (average) of the `TOTAL KCOREMIN` over the number of invocations reflected by the `NUMBER CMDS`.
`MEAN CPU-MIN`	Mean (average) derived from the `NUMBER CMDS` and the `TOTAL CPU-MIN`.
`HOG FACTOR`	Total CPU time divided by elapsed time. Shows the ratio of system availability to system utilization, providing a relative measure of total available CPU time consumed by the process during its execution.
`CHARS TRNSFD`	Total number of characters transferred by the read and write system calls. Might be negative due to overflow.

TABLE 10–5 Daily Command Summary Data *(Continued)*

Column	Description
BLOCKS READ	Total number of the physical block reads and writes that a process performed.

Monthly Command Summary

The format of the Daily Command Summary and the Monthly Command Summary reports are virtually the same. However, the daily summary reports only on the current accounting period while the monthly summary reports on the start of the fiscal period to the current date. In other words, the monthly report is a cumulative summary that reflects the data accumulated since the last invocation of the `monacct` program.

A sample Monthly Command Summary follows.

```
Jan 16 02:30 2004  MONTHLY TOTAL COMMAND SUMMARY Page 1


                                     TOTAL COMMAND SUMMARY
COMMAND   NUMBER      TOTAL   TOTAL     TOTAL   MEAN     MEAN    HOG      CHARS    BLOCKS
NAME        CMDS    KCOREMIN CPU-MIN  REAL-MIN  SIZE-K   CPU-MIN FACTOR  TRNSFD    READ

TOTALS     42718  4398793.50  361.92  956039.00 12154.09 0.01    0.00  16100942848 825171

netscape     789  3110437.25  121.03   79101.12 25699.58 0.15    0.00   3930527232 302486
adeptedi      84  1214419.00   50.20    4174.65 24193.62 0.60    0.01    890216640 107237
acroread     145   165297.78    7.01   18180.74 23566.84 0.05    0.00   1900504064  26053
dtmail         2    64208.90    6.35   20557.14 10112.43 3.17    0.00    250445824  43280
dtaction     800    47602.28   11.26      15.37  4226.93 0.01    0.73    640057536   8095
soffice.      13    35506.79    0.97       9.23 36510.84 0.07    0.11    134754320   5712
dtwm           2    20350.98    3.17   20557.14  6419.87 1.59    0.00    190636032  14049
```

For a description of the data provided by the Monthly Command Summary, see “Daily Command Summary” on page 148.

Last Login Report

This report gives the date when a particular login was last used. You can use this information to find unused logins and login directories that can be archived and deleted. A Last Login Report follows.

```
Jan 16 02:30 2004  LAST LOGIN Page 1
```

```
01-06-12  kryten        01-09-08  protoA      01-10-14  ripley
01-07-14  lister        01-09-08  protoB      01-10-15  scutter1
01-08-16  pmorph        01-10-12  rimmer      01-10-16  scutter2
```

Examining the `pacct` File With `acctcom`

At any time, you can examine the contents of the `/var/adm/pacct`*n* files, or any file with records in the `acct.h` format, by using the `acctcom` program. If you do not specify any files and do not provide any standard input when you run this command, the `acctcom` command reads the `pacct` file. Each record read by the `acctcom` command represents information about a terminated process. Active processes can be examined by running the `ps` command.

The default output of the `acctcom` command provides the following information:

```
# acctcom
COMMAND                            START    END          REAL     CPU    MEAN
NAME        USER     TTYNAME       TIME     TIME       (SECS)  (SECS) SIZE(K)
#accton     root      ?            02:30:01 02:30:01     0.03    0.01  304.00
turnacct    adm       ?            02:30:01 02:30:01     0.42    0.01  320.00
mv          adm       ?            02:30:01 02:30:01     0.07    0.01  504.00
utmp_upd    adm       ?            02:30:01 02:30:01     0.03    0.01  712.00
utmp_upd    adm       ?            02:30:01 02:30:01     0.01    0.01  824.00
utmp_upd    adm       ?            02:30:01 02:30:01     0.01    0.01  912.00
utmp_upd    adm       ?            02:30:01 02:30:01     0.01    0.01  920.00
utmp_upd    adm       ?            02:30:01 02:30:01     0.01    0.01 1136.00
utmp_upd    adm       ?            02:30:01 02:30:01     0.01    0.01  576.00
closewtm    adm       ?            02:30:01 02:30:01     0.10    0.01  664.00
```

Field	Explanation
`COMMAND NAME`	Command name (pound (#) sign if the command was executed with superuser privileges)
`USER`	User name
`TTYNAME`	`tty` name (listed as `?` if unknown)
`START TIME`	Command execution starting time
`END TIME`	Command execution ending time
`REAL (SECS)`	Real time (in seconds)
`CPU (SECS)`	CPU time (in seconds)
`MEAN SIZE (K)`	Mean size (in Kbytes)

You can obtain the following information by using `acctcom` command options.

- State of the `fork/exec` flag (1 for `fork` without `exec`)
- System exit status
- Hog factor
- Total `kcore` minutes
- CPU factor
- Characters transferred
- Blocks read

The following table describes the `acctcom` command options.

TABLE 10–6 Options for the `acctcom` Command

Option	Description
`-a`	Shows average statistics about the processes selected. The statistics are printed after the output is recorded.
`-b`	Reads the files backward, showing latest commands first. This option has no effect if reading standard input.
`-f`	Prints the `fork/exec` flag and system exit status columns. The output is an octal number.
`-h`	Instead of mean memory size, shows the hog factor, which is the fraction of total available CPU time consumed by the process during its execution. Hog factor = *total-CPU-time/elapsed-time*.
`-i`	Prints columns that contains the I/O counts in the output.
`-k`	Shows total `kcore minutes` instead of memory size.
`-m`	Shows mean core size. This size is the default.
`-q`	Prints average statistics, not output records.
`-r`	Shows CPU factor: *user-time*/(*system-time* + *user-time)*.
`-t`	Shows separate system and user CPU times.
`-v`	Excludes column headings from the output.
`-C` *sec*	Shows only processes with total CPU time (system plus user) that exceeds *sec* seconds.
`-e` *time*	Shows processes existing at or before *time*, given in the format *hr[:min[:sec]]*
`-E` *time*	Shows processes starting at or before *time*, given in the format *hr[:min[:sec]]*. Using the same time for both `-S` and `-E`, shows processes that existed at the time.
`-g` *group*	Shows only processes that belong to *group*.

TABLE 10–6 Options for the `acctcom` Command *(Continued)*

Option	Description
`-H` *factor*	Shows only processes that exceed *factor*, where *factor* is the "hog factor" (see the `-h` option).
`-I` *chars*	Shows only processes that transferred more characters than the cutoff number specified by *chars*.
`-l` *line*	Show only processes that belong to the terminal `/dev/`*line*.
`-n` *pattern*	Shows only commands that match *pattern* (a regular expression except that "+" means one or more occurrences).
`-o` *ofile*	Instead of printing the records, copies them in `acct.h` format to *ofile*.
`-O` *sec*	Shows only processes with CPU system time that exceeds *sec* seconds.
`-s` *time*	Show processes existing at or after *time*, given in the format *hr*[:*min*[:*sec*]].
`-S` *time*	Show processes starting at or after *time*, given in the format *hr*[:*min*[:*sec*]].
`-u` *user*	Shows only processes that belong to *user*.

System Accounting Files

The `/var/adm` directory contains the active data collection files. The following table describes the accounting files in this directory.

TABLE 10–7 Files in the `/var/adm` Directory

File	Description
`dtmp`	Output from the `acctdusg` program
`fee`	Output from the `chargefee` program, which are the ASCII `tacct` records
`pacct`	Active process accounting file
`pacct`*n*	Process accounting files that are switched by running the `turnacct` script
`Spacct`*n.MMDD*	Process accounting files for *MMDD* during execution of the `runacct` script

The `/var/adm/acct` directory contains the `nite`, `sum`, and `fiscal` directories. These directories contain the actual data collection files. For example, the `nite` directory contains files that are reused daily by the `runacct` script. A brief summary of the files in the `/var/adm/acct/nite` directory follows.

TABLE 10–8 Files in the `/var/adm/acct/nite` Directory

File	Description
`active`	Used by the `runacct` script to record progress and print warning and error messages
`active.`*MMDD*	Same as the `active` file after the `runacct` script detects an error
`cms`	ASCII total command summary used by the `prdaily` script
`ctacct.`*MMDD*	Connect accounting records in `tacct.h` format
`ctmp`	Output of `acctcon1` program, which consists of connect session records in `ctmp.h` format (`acctcon1` and `acctcon2` are provided for compatibility purposes)
`daycms`	ASCII daily command summary used by the `prdaily` script
`daytacct`	Total accounting records for one day in `tacct.h` format
`disktacct`	Disk accounting records in `tacct.h` format, created by the `dodisk` script
`fd2log`	Diagnostic output during execution of the `runacct` script
`lastdate`	Last day the `runacct` script executed (in `date +%m%d` format)
`lineuse`	`tty` line usage report used by the `prdaily` script
`lock`	Used to control serial use of the `runacct` script
`log`	Diagnostic output from the `acctcon` program
`log.`*MMDD*	Same as the `log` file after the `runacct` script detects an error
`owtmpx`	Previous day's `wtmpx` file
`reboots`	Beginning and ending dates from the `wtmpx` file, and a listing of reboots
`statefile`	Used to record current state during execution of the `runacct` script
`tmpwtmp`	`wtmpx` file corrected by the `wtmpfix` program
`wtmperror`	Contains `wtmpfix` error messages
`wtmperror` *MMDD*	Same as the `wtmperror` file after the `runacct` script detects an error
`wtmp`*MMDD*	The `runacct` script's copy of the `wtmpx` file

The `sum` directory contains the cumulative summary files updated by the `runacct` script and used by the `monacct` script. The following table summarizes the files in the `/var/adm/acct/sum` directory.

TABLE 10–9 Files in the `/var/adm/acct/sum` Directory

File	Description
`cms`	Total command summary file for current fiscal period in binary format
`cmsprev`	Command summary file without latest update
`daycms`	Command summary file for the day's usage in internal summary format
`loginlog`	Record of last date each user logged in; created by the `lastlogin` script and used in the `prdaily` script
`rprt.`*MMDD*	Saved output of `prdaily` script
`tacct`	Cumulative total accounting file for current fiscal period
`tacctprev`	Same as the `tacct` file without latest update
`tacct.`*MMDD*	Total accounting file for *MMDD*

The fiscal directory contains periodic summary files that are created by the `monacct` script. The following table summarizes the files in the `/var/adm/acct/fiscal` directory.

TABLE 10–10 Files in the `/var/adm/acct/fiscal` Directory

File	Description
`cms`*n*	Total command summary file for fiscal period *n* in internal summary format
`fiscrpt`*n*	Report similar to `rprt`*n* for fiscal period *n*
`tacct`*n*	Total accounting file for fiscal period *n*

Files Produced by the `runacct` Script

The following table summarizes the most useful files produced by the `runacct` script. These files are found in the `/var/adm/acct` directory.

TABLE 10–11 Files Created by the `runacct` Script

File	Description
`nite/daytacct`	The total accounting file for the day in `tacct.h` format.

TABLE 10–11 Files Created by the `runacct` Script *(Continued)*

File	Description
`nite/lineuse`	The `runacct` script calls the `acctcon` program to gather data on terminal line usage from the `/var/adm/acct/nite/tmpwtmp` file and writes the data to the `/var/adm/acct/nite/lineuse` file. The `prdaily` script uses this data to report line usage. This report is especially useful for detecting bad lines. If the ratio between the number of logouts to logins is greater than three to one, the line is very likely failing.
`sum/cms`	This file is the accumulation of each day's command summaries. The accumulation restarts when the `monacct` script is executed. The ASCII version is the `nite/cms` file.
`sum/daycms`	The `runacct` script calls the `acctcms` program to process the commands used during the day to create the Daily Command Summary report and stores the data in the `/var/adm/acct/sum/daycms` file. The ASCII version is the `/var/adm/acct/nite/daycms` file.
`sum/loginlog`	The `runacct` script calls the `lastlogin` script to update the last date logged in for the logins in the `/var/adm/acct/sum/loginlog` file. The `lastlogin` command also removes from this file any logins that are no longer valid.
`sum/rprt.`*MMDD*	Each execution of the `runacct` script saves a copy of the daily report that was printed by the `prdaily` script.
`sum/tacct`	Contains the accumulation of each day's `nite/daytacct` data and is used for billing purposes. The `monacct` script restarts accumulating this data each month or fiscal period.

CHAPTER 11

Managing System Performance (Overview)

Achieving good performance from a computer or network is an important part of system administration. This chapter provides an overview of some factors that contribute to managing the performance of the computer systems in your care.

This is a list of the overview information in this chapter.

- "What's New in Managing System Performance?" on page 157
- "Where to Find System Performance Tasks" on page 158
- "System Performance and System Resources" on page 159
- "Processes and System Performance" on page 159
- "About Monitoring System Performance" on page 161

What's New in Managing System Performance?

This section describes new or changed features in managing system performance in the Solaris release. For a complete listing of new Solaris features and a description of Solaris releases, see the *Solaris 10 What's New*.

Enhanced `pfiles` Tool

Solaris 10: The `/proc` file system has been enhanced to include file name information in the `/proc/`*pic*`/path` directory. This information is used by `pfiles` to display file names for each file in the process. This change provides new insight into process behavior. For more information, see "How to Display Information About Processes" on page 169 and the `proc(1)` man page.

CPU Performance Counters

Solaris 10: The CPU Performance Counter (CPC) system has been updated to give better access to the performance analysis features available in the SPARC and x86 platforms that run the Solaris Operating System.

The CPC commands cpustat and cputrack have enhanced, command-line syntax for specifying CPU information. For example, in previous versions of the Solaris OS, you were required to specify two counters. The configuration of both commands now allows you to specify only one counter, as shown in the following example:

```
# cputrack -c pic0=Cycle_cnt ls -d .
time lwp      event      pic0      pic1
.
0.034   1       exit    841167
```

For simple measurements, you can even omit the counter configuration, as shown in the following example:

```
# cputrack -c Cycle_cnt ls -d .
   time lwp      event      pic0      pic1
  .
  0.016   1       exit    850736
```

For more information on using the cpustat command, see the cpustat(1M) man page. For more information on using the cputrack command, see the cputrack(1) man page.

Where to Find System Performance Tasks

System Performance Task	For More Information
Manage processes	Chapter 12, “Managing System Processes (Tasks)”
Monitor system performance	Chapter 13, “Monitoring System Performance (Tasks)”
Change Solaris tunable parameters	*Solaris Tunable Parameters Reference Manual*
Manage System Performance Tasks	Chapter 2, “Projects and Tasks (Overview),” in *System Administration Guide: Solaris Containers-Resource Management and Solaris Zones*
Manage Processes With FX and FS Schedulers	Chapter 8, “Fair Share Scheduler (Overview),” in *System Administration Guide: Solaris Containers-Resource Management and Solaris Zones*

System Performance and System Resources

The performance of a computer system depends upon how the system uses and allocates its resources. Monitor your system's performance regularly so that you know how it behaves under normal conditions. You should have a good idea of what to expect, and be able to recognize a problem when it occurs.

System resources that affect performance are described in the following table.

System Resource	Description
Central processing unit (CPU)	The CPU processes instructions by fetching instructions from memory and executing them.
Input/output (I/O) devices	I/O devices transfer information into and out of the computer. Such a device could be a terminal and keyboard, a disk drive, or a printer.
Memory	Physical (or main) memory is the amount of random access memory (RAM) on the system.

Chapter 13, “Monitoring System Performance (Tasks),” describes the tools that display statistics about the system's activity and performance.

Processes and System Performance

The following table describes terms that are related to processes.

TABLE 11–1 Process Terminology

Term	Description
Process	Any system activity or job. Each time you boot a system, execute a command, or start an application, the system activates one or more processes.
Lightweight process (LWP)	A virtual CPU or execution resource. LWPs are scheduled by the kernel to use available CPU resources based on their scheduling class and priority. LWPs include a kernel thread and an LWP. A kernel thread contains information that has to be in memory all the time. An LWP contains information that is swappable.
Application thread	A series of instructions with a separate stack that can execute independently in a user's address space. Application threads can be multiplexed on top of LWPs.

A process can consist of multiple LWPs and multiple application threads. The kernel schedules a kernel-thread structure, which is the scheduling entity in the SunOS environment. Various process structures are described in the following table.

TABLE 11–2 Process Structures

Structure	Description
`proc`	Contains information that pertains to the whole process and must be in main memory all the time
`kthread`	Contains information that pertains to one LWP and must be in main memory all the time
`user`	Contains the "per process" information that is swappable
`klwp`	Contains the "per LWP process" information that is swappable

The following figure illustrates the relationships among these process structures.

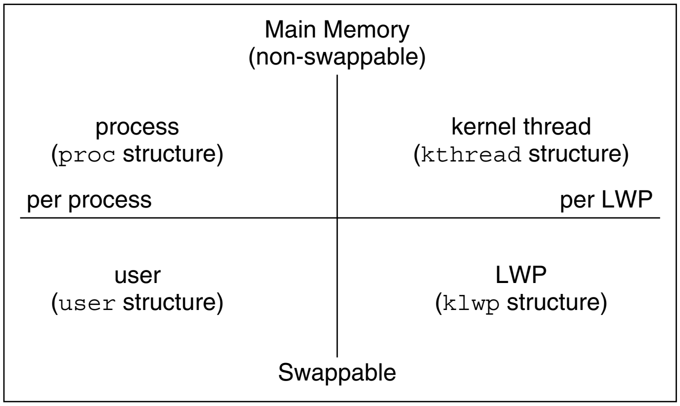

FIGURE 11–1 Relationships Among Process Structures

Most process resources are accessible to all the threads in the process. Almost all process virtual memory is shared. A change in shared data by one thread is available to the other threads in the process.

About Monitoring System Performance

While your computer is running, counters in the operating system are incremented to track various system activities.

System activities that are tracked are as follows:

- Central processing unit (CPU) utilization
- Buffer usage
- Disk and tape input/output (I/O) activity
- Terminal device activity
- System call activity
- Context switching
- File access
- Queue activity
- Kernel tables
- Interprocess communication
- Paging
- Free memory and swap space
- Kernel memory allocation (KMA)

Monitoring Tools

The Solaris software provides several tools to help you track how your system is performing. The following table describes these tools.

TABLE 11–3 Performance Monitoring Tools

Command	Description	For More Information
`cpustat` and `cputrack` commands	Monitors performance of a system or a process using CPU performance counters.	`cpustat`(1M) and `cputrack`(1)
`netstat` and `nfsstat` commands	Displays information about network performance	`netstat`(1M) and `nfsstat`(1M)
`ps` and `prstat` commands	Displays information about active processes	Chapter 12, "Managing System Processes (Tasks)"
`sar` and `sadc` commands	Collects and reports on system activity data	Chapter 13, "Monitoring System Performance (Tasks)"
Sun Enterprise SyMON	Collects system activity data on Sun's enterprise-level systems	*Sun Enterprise SyMON 2.0.1 Software User's Guide*

TABLE 11–3 Performance Monitoring Tools *(Continued)*

Command	Description	For More Information
`swap` command	Displays information about available swap space on your system	Chapter 20, "Configuring Additional Swap Space (Tasks)," in *System Administration Guide: Devices and File Systems*
`vmstat` and `iostat` commands	Summarizes system activity data, such as virtual memory statistics, disk usage, and CPU activity	Chapter 13, "Monitoring System Performance (Tasks)"
`cputrack` and `cpustat` commands	Assists in accessing hardware performance counter facilities provided by microprocessors	`cputrack`(1) and `cpustat`(1M) man pages
`kstat` and `mpstat` commands	Examines the available kernel statistics, or `kstats`, on the system and reports those statistics which match the criteria specified on the command line. The `mpstat` command reports processor statistics in tabular form.	`kstat`(1M) and `mpstat`(1M) man pages.

CHAPTER 12

Managing System Processes (Tasks)

This chapter describes the procedures for managing system processes.

For information on the procedures associated with managing system processes, see the following:

- "Managing System Processes (Task Map)" on page 163
- "Managing Process Class Information (Task Map)" on page 174

For overview information about managing system processes, see the following:

- "Commands for Managing System Processes" on page 164
- "Managing Process Class Information" on page 174

Managing System Processes (Task Map)

Task	Description	For Instructions
List processes.	Use the `ps` command to list all the processes on a system.	"How to List Processes" on page 167
Display information about processes.	Use the `pgrep` command to obtain the process IDs for processes that you want to display more information about.	"How to Display Information About Processes" on page 169
Control processes.	Locate processes by using the `pgrep` command. Then, use the appropriate `pcommand` (`/proc`) to control the process. See Table 12–3 for a description of the (`/proc`) commands.	"How to Control Processes" on page 170

Task	Description	For Instructions
Kill a process.	Locate a process, either by process name or process ID. You can use either the `pkill` or `kill` commands to terminate the process.	"How to Terminate a Process (`pkill`)" on page 171 "How to Terminate a Process (`kill`)" on page 172

Commands for Managing System Processes

The following table describes the commands for managing system processes.

TABLE 12–1 Commands for Managing Processes

Command	Description	Man Page
`ps`, `pgrep`, `prstat`, `pkill`	Checks the status of active processes on a system, as well as displays detailed information about the processes	`ps`(1), `pgrep`(1), and`prstat`(1M)
`pkill`	Functions identically to `pgrep` but finds or signals processes by name or other attribute and terminates the process. Each matching process is signaled as if by the `kill` command, instead of having its process ID printed.	`pgrep`(1), and `pkill`(1) `kill`(1)
`pargs, preap`	Assists with processes debugging	`pargs`(1), and `preap`(1)
`dispadmin`	Lists default process scheduling policies	`dispadmin`(1M)
`priocntl`	Assigns processes to a priority class and manages process priortities	`priocntl`(1)
`nice`	Changes the priority of a timesharing process	`nice`(1)
`psrset`	Binds specific process groups to a group of processors rather than to just a single processor	`psrset`(1M)

The Solaris Management Console's Processes tool enables you to manage processes with a user-friendly interface. For information on using and starting the Solaris Management Console, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

Using the ps Command

The ps command enables you to check the status of active processes on a system, as well as display technical information about the processes. This data is useful for administrative tasks such as determining how to set process priorities.

Depending on which options you use, the ps command reports the following information:

- Current status of the process
- Process ID
- Parent process ID
- User ID
- Scheduling class
- Priority
- Address of the process
- Memory used
- CPU time used

The following table describes some fields that are reported by the ps command. Which fields are displayed depend on which option you choose. For a description of all available options, see the ps(1) man page.

TABLE 12–2 Summary of Fields in ps Reports

Field	Description
UID	The effective user ID of the process's owner.
PID	The process ID.
PPID	The parent process ID.
C	The processor xutilization for scheduling. This field is not displayed when the -c option is used.
CLS	The scheduling class to which the process belongs such as real-time, system, or timesharing. This field is included only with the -c option.
PRI	The kernel thread's scheduling priority. Higher numbers indicate a higher priority.
NI	The process's nice number, which contributes to its scheduling priority. Making a process “nicer” means lowering its priority.
ADDR	The address of the proc structure.
SZ	The virtual address size of the process.
WCHAN	The address of an event or lock for which the process is sleeping.

TABLE 12–2 Summary of Fields in ps Reports *(Continued)*

Field	Description
STIME	The starting time of the process in hours, minutes, and seconds.
TTY	The terminal from which the process, or its parent, was started. A question mark indicates that there is no controlling terminal.
TIME	The total amount of CPU time used by the process since it began.
CMD	The command that generated the process.

Using the /proc File System and Commands

You can display detailed information about the processes that are listed in the /proc directory by using process commands. The following table lists the /proc process commands. The /proc directory is also known as the process file system (PROCFS). Images of active processes are stored here by their process ID number.

TABLE 12–3 Process Commands (/proc)

Process Command	Description
pcred	Displays process credential information
pfiles	Reports fstat and fcntl information for open files in a process
pflags	Prints /proc tracing flags, pending signals and held signals, and other status information
pldd	Lists the dynamic libraries that are linked into a process
pmap	Prints the address space map of each process
psig	Lists the signal actions and handlers of each process
prun	Starts each process
pstack	Prints a hex+symbolic stack trace for each lwp in each process
pstop	Stops each process
ptime	Times a process by using microstate accounting
ptree	Displays the process trees that contain the process
pwait	Displays status information after a process terminates
pwdx	Displays the current working directory for a process

For more information, see proc(1).

The process tools are similar to some options of the ps command, except that the output that is provided by these commands is more detailed.

In general, the process commands do the following:

- Display more information about processes, such as fstat and fcntl, working directories, and trees of parent and child processes
- Provide control over processes by allowing users to stop or resume them

Managing Processes With Process Commands (/proc)

You can display detailed, technical information about processes or control active processes by using some of the process commands. Table 12–3 lists some of the /proc commands.

If a process becomes trapped in an endless loop, or if the process takes too long to execute, you might want to stop (kill) the process. For more information about stopping processes using the kill or the pkill command, see Chapter 12, “Managing System Processes (Tasks).”

The /proc file system is a directory hierarchy that contains additional subdirectories for state information and control functions.

The /proc file system also provides an xwatchpoint facility that is used to remap read-and-write permissions on the individual pages of a process's address space. This facility has no restrictions and is MT-safe.

Debugging tools have been modified to use /proc's xwatchpoint facility, which means that the entire xwatchpoint process is faster.

The following restrictions have been removed when you set xwatchpoints by using the dbx debugging tool:

- Setting xwatchpoints on local variables on the stack due to SPARC based system register windows
- Setting xwatchpoints on multithreaded processes

For more information, see the proc(4), and mdb(1) man pages.

▼ How to List Processes

● **Use the ps command to list all the processes on a system.**

```
$ ps [-efc]
```

ps Displays only the processes that are associated with your login session.

-ef Displays full information about all the processes that are being executed on the system.

-c Displays process scheduler information.

Example 12–1 Listing Processes

The following example shows output from the ps command when no options are used.

```
$ ps
  PID TTY      TIME COMD
 1664 pts/4    0:06 csh
 2081 pts/4    0:00 ps
```

The following example shows output from the ps -ef command. This output shows that the first process that is executed when the system boots is sched (the swapper) followed by the init process, pageout, and so on.

```
$ ps -ef
     UID   PID  PPID  C    STIME TTY      TIME CMD
    root     0     0  0   Dec 20 ?        0:17 sched
    root     1     0  0   Dec 20 ?        0:00 /etc/init -
    root     2     0  0   Dec 20 ?        0:00 pageout
    root     3     0  0   Dec 20 ?        4:20 fsflush
    root   374   367  0   Dec 20 ?        0:00 /usr/lib/saf/ttymon
    root   367     1  0   Dec 20 ?        0:00 /usr/lib/saf/sac -t 300
    root   126     1  0   Dec 20 ?        0:00 /usr/sbin/rpcbind
    root    54     1  0   Dec 20 ?        0:00 /usr/lib/sysevent/syseventd
    root    59     1  0   Dec 20 ?        0:00 /usr/lib/picl/picld
    root   178     1  0   Dec 20 ?        0:03 /usr/lib/autofs/automountd
    root   129     1  0   Dec 20 ?        0:00 /usr/sbin/keyserv
    root   213     1  0   Dec 20 ?        0:00 /usr/lib/lpsched
    root   154     1  0   Dec 20 ?        0:00 /usr/sbin/inetd -s
    root   139     1  0   Dec 20 ?        0:00 /usr/lib/netsvc/yp/ypbind ...
    root   191     1  0   Dec 20 ?        0:00 /usr/sbin/syslogd
    root   208     1  0   Dec 20 ?        0:02 /usr/sbin/nscd
    root   193     1  0   Dec 20 ?        0:00 /usr/sbin/cron
    root   174     1  0   Dec 20 ?        0:00 /usr/lib/nfs/lockd
  daemon   175     1  0   Dec 20 ?        0:00 /usr/lib/nfs/statd
    root   376     1  0   Dec 20 ?        0:00 /usr/lib/ssh/sshd
    root   226     1  0   Dec 20 ?        0:00 /usr/lib/power/powerd
    root   315     1  0   Dec 20 ?        0:00 /usr/lib/nfs/mountd
    root   237     1  0   Dec 20 ?        0:00 /usr/lib/utmpd
    .
    .
    .
```

▼ How to Display Information About Processes

1 Obtain the process ID of the process that you want to display more information about.

```
# pgrep process
```

where *process* is the name of the process you want to display more information about.

The process ID is displayed in the first column of the output.

2 Display the process information that you need.

```
# /usr/bin/pcommand pid
```

pcommand Is the (`/proc`) command that you want to run. Table 12–3 lists and describes these commands.

pid Identifies the process ID.

Example 12–2 Displaying Information About Processes

The following example shows how to use process commands to display more information about a `cron` process.

```
# pgrep cron     1
4780
# pwdx 4780     2
4780:   /var/spool/cron/atjobs
# ptree 4780     3
4780  /usr/sbin/cron
# pfiles 4780     4
4780:   /usr/sbin/cron
  Current rlimit: 256 file descriptors
   0: S_IFCHR mode:0666 dev:290,0 ino:6815752 uid:0 gid:3 rdev:13,2
      O_RDONLY|O_LARGEFILE
      /devices/pseudo/mm@0:null
   1: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
      O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
      /var/cron/log
   2: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
      O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
      /var/cron/log
   3: S_IFIFO mode:0600 dev:32,128 ino:42049 uid:0 gid:0 size:0
      O_RDWR|O_LARGEFILE
      /etc/cron.d/FIFO
   4: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
      O_RDWR|O_NONBLOCK
   5: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
      O_RDWR
```

1. Obtains the process ID for the `cron` process
2. Displays the current working directory for the `cron` process
3. Displays the process tree that contains the `cron` process
4. Displays `fstat` and `fcntl` information

▼ How to Control Processes

1 Obtain the process ID of the process that you want to control.

```
# pgrep process
```

where *process* is the name of the process you want to control.

The process ID displayed in the first column of the output.

2 Use the appropriate process command to control the process.

```
# /usr/bin/pcommand pid
```

pcommand Is the process (`/proc`) command that you want to run. Table 12–3 lists and describes these commands.

pid Identifies the process ID.

3 Verify the process status.

```
# ps -ef | grep pid
```

Example 12–3 Controlling Processes

The following example shows how to use process command to stop and restart the `dtpad` process.

```
# pgrep dtpad       1
2921
# pstop 2921      2
# prun 2921       3
```

1. Obtains the process ID for the `dtpad` process
2. Stops the `dtpad` process
3. Restarts the `dtpad` process

Terminating a Process (pkill, kill)

Sometimes, you might need to stop (kill) a process. The process might be in an endless loop. Or, you might have started a large job that you want to stop before it is completed. You can kill any process that you own. Superuser can kill any process in the system except for those processes with process IDs of 0, 1, 2, 3, and 4. Killing these processes most likely will crash the system.

For more information, see the pgrep(1) and pkill(1) and kill(1) man pages.

▼ How to Terminate a Process (pkill)

1. **(Optional) Become superuser or assume an equivalent role to terminate the process of another user.**

2. **Obtain the process ID for the process that you want to terminate.**

   ```
   $ pgrep process
   ```

   where *process* is the name of the process that you want to terminate.

   For example:

   ```
   $ pgrep netscape
   587
   566
   ```

   The process ID is displayed in the output.

   Note – To obtain process information on a Sun Ray™, use the following commands:

   ```
   # ps -fu user
   ```

   This command lists all user processes.

   ```
   # ps -fu user | grep process
   ```

   This command locates a specific process for a user.

   ---

3. **Terminate the process.**

   ```
   $ pkill [signal] process
   ```

   signal When no signal is included in the pkill command-line syntax, the default signal that is used is –15 (SIGKILL). Using the –9 signal (SIGTERM) with the pkill command ensures that the process terminates promptly. However, the –9 signal should not be used to kill certain processes, such as a database process, or an LDAP server process. The result is that data might be lost.

process Is the name of the process to stop.

Tip – When using the `pkill` command to terminate a process, first try using the command by itself, without including a signal option. Wait a few minutes to see if the process terminates before using the `pkill` command with the -9 signal.

---

4 **Verify that the process has been terminated.**

```
$ pgrep process
```

The process you terminated should no longer be listed in the output of the `pgrep` command.

▼ How to Terminate a Process (`kill`)

1 **(Optional) Become superuser or assume an equivalent role to terminate the process of another user.**

2 **Obtain the process ID of the process that you want to terminate.**

```
$ ps -fu user
```

where *user* is the user that you want to display processes for.

The process ID is displayed in the first column of the output.

3 **Terminate the process.**

```
$ kill [signal-number] pid
```

signal When no signal is included in the `kill` command-line syntax, the default signal that is used is –15 (`SIGKILL`). Using the –9 signal (`SIGTERM`) with the `kill` command ensures that the process terminates promptly. However, the –9 signal should not be used to kill certain processes, such as a database process, or an LDAP server process. The result is that data might be lost.

pid Is the process ID of the process that you want to terminate.

Tip – When using the `kill` command to stop a process, first try using the command by itself, without including a signal option. Wait a few minutes to see if the process terminates before using the `kill` command with the -9 signal.

---

4 **Verify that the process has been terminated.**

```
$ pgrep pid
```

The process you terminated should no longer be listed in the output of the pgrep command.

Debugging a Process (pargs, preap)

The pargs command and the preap command improve process debugging. The pargs command prints the arguments and environment variables associated with a live process or core file. The preap command removes defunct (zombie) processes. A zombie process has not yet had its exit status claimed by its parent. These processes are generally harmless but can consume system resources if they are numerous. You can use the pargs and preap commands to examine any process that you have the privileges to examine. As superuser, you can examine any process.

For information on using the preap command, see the preap(1) man page. For information on the using the pargs command, see the pargs(1) man page. See also, the proc(1) man page.

EXAMPLE 12–4 Debugging a Process (pargs)

The pargs command solves a long-standing problem of being unable to display with the ps command all the arguments that are passed to a process. The following example shows how to use the pargs command in combination with the pgrep command to display the arguments that are passed to a process.

```
# pargs ‘pgrep ttymon‘
579:    /usr/lib/saf/ttymon -g -h -p system-name console login:
-T sun -d /dev/console -l
argv[0]: /usr/lib/saf/ttymon
argv[1]: -g
argv[2]: -h
argv[3]: -p
argv[4]: system-name console login:
argv[5]: -T
argv[6]: sun
argv[7]: -d
argv[8]: /dev/console
argv[9]: -l
argv[10]: console
argv[11]: -m
argv[12]: ldterm,ttcompat
548:    /usr/lib/saf/ttymon
argv[0]: /usr/lib/saf/ttymon
```

The following example shows how to use the pargs -e command to display the environment variables that are associated with a process.

EXAMPLE 12–4 Debugging a Process (pargs) *(Continued)*

```
$ pargs -e 6763
6763: tcsh
envp[0]: DISPLAY=:0.0
```

Managing Process Class Information (Task Map)

Task	Description	For Instructions
Display basic information about process classes.	Use the `priocntl -l` command. to Display process scheduling classes and priority ranges.	"How to Display Basic Information About Process Classes (`priocntl`)" on page 175
Display the global priority of a process.	Use the `ps -ecl` command to display the global priority of a process.	"How to Display the Global Priority of a Process" on page 176
Designate a process priority.	Start a process with a designated priority by using the `priocntl -e -c command`.	"How to Designate a Process Priority (`priocntl`)" on page 176
Change scheduling parameters of a timesharing process.	Use the `priocntl -s -m` command to change scheduling parameters in a timesharing process.	"How to Change Scheduling Parameters of a Timesharing Process (`priocntl`)" on page 177
Change the class of a process.	Use the `priocntl -s -c` command to change the class of a process.	"How to Change the Class of a Process (`priocntl`)" on page 178
Change the priority of a process.	Use the `/usr/bin/nice` command with the appropriate options to lower or raise the priority of a process.	"How to Change the Priority of a Process (`nice`)" on page 179

Managing Process Class Information

The following list identifies the process scheduling classes that can be configured on your system. Also included is the user priority range for the timesharing class.

The possible process scheduling classes are as follows:

- Fair share (`FSS`)
- Fixed (`FX`)
- System (`SYS`)
- Interactive (`IA`)

- Real-time (RT)
- Timesharing (TS)
  - The user-supplied priority ranges from -60 to +60.
  - The priority of a process is inherited from the parent process. This priority is referred to as the *user-mode priority*.
  - The system looks up the user-mode priority in the timesharing dispatch parameter table. Then, the system adds in any `nice` or `priocntl` (user-supplied) priority and ensures a 0–59 range to create a *global priority*.

Changing the Scheduling Priority of Processes (`priocntl`)

The scheduling priority of a process is the priority assigned by the process scheduler, according to scheduling policies. The `dispadmin` command lists the default scheduling policies. For more information, see the `dispadmin`(1M) man page.

You can use the `priocntl` command to assign processes to a priority class and to manage process priorities. For instructions on using the `priocntl` command to manage processes, see "How to Designate a Process Priority (`priocntl`)" on page 176.

▼ How to Display Basic Information About Process Classes (`priocntl`)

● **Display process scheduling classes and priority ranges with the `priocntl -l` command.**

```
$ priocntl -l
```

Example 12–5 Displaying Basic Information About Process Classes (`priocntl`)

The following example shows output from the `priocntl -l` command.

```
# priocntl -l
CONFIGURED CLASSES
==================

SYS (System Class)

TS (Time Sharing)
        Configured TS User Priority Range: -60 through 60

FX (Fixed priority)
```

```
        Configured FX User Priority Range: 0 through 60

IA (Interactive)
        Configured IA User Priority Range: -60 through 60
```

How to Display the Global Priority of a Process

- **Display the global priority of a process by using the `ps` command.**

  ```
  $ ps -ecl
  ```

  The global priority is listed under the `PRI` column.

Example 12–6 Displaying the Global Priority of a Process

The following example shows `ps -ecl` command output. The values in the `PRI` column show that the `pageout` process has the highest priority, while the `sh` process has the lowest priority.

```
$ ps -ecl
 F S UID PID  PPID CLS PRI  ADDR      SZ  WCHAN    TTY      TIME   COMD
19 T 0   0    0    SYS 96   f00d05a8   0           ?        0:03  sched
 8 S 0   1    0    TS  50   ff0f4678 185  ff0f4848 ?       36:51   init
19 S 0   2    0    SYS 98   ff0f4018   0  f00c645c ?        0:01 pageout
19 S 0   3    0    SYS 60   ff0f5998   0  f00d0c68 ?      241:01 fsflush
 8 S 0   269  1    TS  58   ff0f5338 303  ff49837e ?        0:07    sac
 8 S 0   204  1    TS  43   ff2f6008  50  ff2f606e console  0:02     sh
```

How to Designate a Process Priority (`priocntl`)

1. **(Optional) Assume the Primary Administrator role, or become superuser.**

   The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2. **Start a process with a designated priority.**

   # priocntl -e -c *class* **-m** *user-limit* **-p** *pri command-name*

   `-e` Executes the command.

   `-c` *class* Specifies the class within which to run the process. The valid classes are `TS` (timesharing), `RT` (real time), IA (interactive), FSS (fair share), and FX (fixed priority).

   `-m` *user-limit* When you use the `-p` option, specifies the maximum amount you can raise or lower your priority,

`-p` *pri command-name* — Lets you specify the relative priority in the RT class for a real-time thread. For a timesharing process, the `-p` option lets you specify the user-supplied priority, which ranges from -60 to +60.

3. **Verify the process status.**

```
# ps -ecl | grep command-name
```

Example 12–7 Designating a Process Priority (`priocntl`)

The following example shows how to start the `find` command with the highest possible user-supplied priority.

```
# priocntl -e -c TS -m 60 -p 60 find . -name core -print
# ps -ecl | grep find
```

▼ How to Change Scheduling Parameters of a Timesharing Process (`priocntl`)

1. **(Optional) Assume the Primary Administrator role, or become superuser.**

   The Primary Administrator role includes the Primary Administrator profile. To create the role and assign the role to a user, see Chapter 2, "Working With the Solaris Management Console (Tasks)," in *System Administration Guide: Basic Administration*.

2. **Change the scheduling parameters of a running timesharing process.**

```
# priocntl -s -m user-limit [-p user-priority] -i idtype idlist
```

`-s` — Lets you set the upper limit on the user priority range and change the current priority.

`-m` *user-limit* — When you use the `-p` option, specifies the maximum amount you can raise or lower the priority.

`-p` *user-priority* — Allows you to designate a priority.

`-i` *xidtype xidlist* — Uses a combination of *xidtype* and *xidlist* to identify the process or processes. The *xidtype* specifies the type of ID, such as the process ID or the user ID. Use *xidlist* to identify a list of process IDs or user IDs.

3. **Verify the process status.**

```
# ps -ecl | grep idlist
```

Example 12–8 Changing Scheduling Parameters of a Timesharing Process (`priocntl`)

The following example shows how to execute a command with a 500-millisecond time slice, a priority of 20 in the RT class, and a global priority of 120.

```
# priocntl -e -c RT -m 500 -p 20 myprog
# ps -ecl | grep myprog
```

▼ How to Change the Class of a Process (`priocntl`)

1 **(Optional) Become superuser or assume an equivalent role.**

2 **Change the class of a process.**

```
# priocntl -s -c class -i idtype  idlist
```

`-s` Lets you set the upper limit on the user priority range and change the current priority.

`-c` *class* Specifies the class, `TS` for time-sharing or `RT for real-time`, to which you are changing the process.

`-i` *idtype idlist* Uses a combination of *xidtype* and *xidlist* to identify the process or processes. The *xidtype* specifies the type of ID, such as the process ID or user ID. Use *xidlist* to identify a list of process IDs or user IDs.

Note – You must be superuser or working in a real-time shell to change a process from, or to, a real-time process. If, as superuser, you change a user process to the real-time class, the user cannot subsequently change the real-time scheduling parameters by using the `priocntl -s` command.

---

3 **Verify the process status.**

```
# ps -ecl | grep idlist
```

Example 12–9 Changing the Class of a Process (`priocntl`)

The following example shows how to change all the processes that belong to user `15249` to real-time processes.

```
# priocntl -s -c RT -i uid 15249
# ps -ecl | grep 15249
```

Changing the Priority of a Timesharing Process (`nice`)

The `nice` command is only supported for backward compatibility to previous Solaris releases. The `priocntl` command provides more flexibility in managing processes.

The priority of a process is determined by the policies of its scheduling class and by its *nice number*. Each timesharing process has a global priority. The global priority is calculated by adding the user-supplied priority, which can be influenced by the `nice` or `priocntl` commands, and the system-calculated priority.

The execution priority number of a process is assigned by the operating system. The priority number is determined by several factors, including the process's scheduling class, how much CPU time it has used, and in the case of a timesharing process, its `nice` number.

Each timesharing process starts with a default `nice` number, which it inherits from its parent process. The `nice` number is shown in the `NI` column of the `ps` report.

A user can lower the priority of a process by increasing its user-supplied priority. However, only superuser can lower a `nice` number to increase the priority of a process. This restriction prevents users from increasing the priorities of their own processes, thereby monopolizing a greater share of the CPU.

The `nice` numbers range from 0 to +39, with 0 representing the highest priority. The default `nice` value for each timesharing process is 20. Two versions of the command are available: the standard version, `/usr/bin/nice`, and the C shell built-in command.

▼ How to Change the Priority of a Process (`nice`)

Using this procedure, a user can lower the priority of a process. However, superuser can raise or lower the priority of a process.

Note – This section describes the syntax of the `/usr/bin/nice` command and not the C-shell built-in `nice`command. For information about the C-shell `nice` command, see the `csh(1)` man page.

---

1. **Determine whether you want to change the priority of a process, either as a user or as superuser. Then, select one of the following:**

   - **As a user, follow the examples in Step 2 to lower the priority of a command.**
   - **As a superuser, follow the examples in Step 3 to raise or lower priorities of a command.**

2 **As a user, lower the priority of a command by increasing the `nice` number.**

The following `nice` command executes *command-name* with a lower priority by raising the nice number by 5 units.

```
$ /usr/bin/nice -5 command-name
```

In the preceding command, the minus sign designates that what follows is an option. This command could also be specified as follows:

```
% /usr/bin/nice -n 5 command-name
```

The following `nice` command lowers the priority of *command-name* by raising the nice number by the default increment of 10 units, but not beyond the maximum value of 39.

```
% /usr/bin/nice command-name
```

3 **As superuser or assuming an equivalent role, raise or lower the priority of a command by changing the `nice` number.**

The following `nice` command raises the priority of *command-name* by lowering the nice number by 10 units, but not below the minimum value of 0.

```
# /usr/bin/nice --10 command-name
```

In the preceding command, the first minus sign designates that what follows is an option. The second minus sign indicates a negative number.

The following `nice` command lowers the priority of *command-name* by raising the nice number by 5 units, but not beyond the maximum value of 39.

```
# /usr/bin/nice -5 command-name
```

See Also For more information, see the `nice(1)` man page.

Troubleshooting Problems With System Processes

Here are some tips on obvious problems you might encounter:

- Look for several identical jobs that are owned by the same user. This problem might occur because of a running script that starts a lot of background jobs without waiting for any of the jobs to finish.
- Look for a process that has accumulated a large amount of CPU time. You can identify this problem by checking the `TIME` field in the `ps` output. Possibly, the process is in an endless loop.

- Look for a process that is running with a priority that is too high. Use the `ps -c` command to check the `CLS` field, which displays the scheduling class of each process. A process executing as a real-time (`RT`) process can monopolize the CPU. Or, look for a timesharing (`TS`) process with a high `nice` number. A user with superuser privileges might have increased the priority of a process. The system administrator can lower the priority by using the `nice` command.
- Look for a runaway process. A runaway process progressively uses more and more CPU time. You can identify this problem by looking at the time when the process started (`STIME`) and by watching the cumulation of CPU time (`TIME`) for a while.

◆ ◆ ◆ CHAPTER 13

Monitoring System Performance (Tasks)

This chapter describes procedures for monitoring system performance by using the `vmstat`, `iostat`, `df`, and `sar` commands.

For information on the procedures that are associated with monitoring system performance, see the following:

- "Displaying System Performance Information (Task Map)" on page 183
- "Monitoring System Activities (Task Map)" on page 191

Displaying System Performance Information (Task Map)

Task	Description	For Instructions
Display virtual memory Statistics.	Collect virtual memory statistics by using the `vmstat` command.	"How to Display Virtual Memory Statistics (`vmstat`)" on page 185
Display system event information.	Display system event information by using the `vmstat` command with the `-s` option	"How to Display System Event Information (`vmstat -s`)" on page 186
Display swapping statistics.	Use the `vmstat` command with the `-S` option to display swapping statistics.	"How to Display Swapping Statistics (`vmstat -S`)" on page 187
Display interrupts per device.	Use the `vmstat` command with the `-i` option to show the number of interrupts per device.	"How to Display Interrupts Per Device (`vmstat -i`)" on page 187
Display disk utilization.	Use the `iostat` command to report disk input and output statistics.	"How to Display Disk Utilization Information (`iostat`)" on page 188

Task	Description	For Instructions
Display extended disk statistics.	Use the iostat command with the -xtcoption to display extended disk statistics.	"How to Display Extended Disk Statistics (iostat -xtc)" on page 189
Display disk space information.	The df -k command displays disk space information in Kbytes.	"How to Display Disk Space Information (df -k)" on page 190

Displaying Virtual Memory Statistics (vmstat)

You can use the vmstat command to report virtual memory statistics and information about system events such as CPU load, paging, number of context switches, device interrupts, and system calls. The vmstat command can also display statistics on swapping, cache flushing, and interrupts.

The following table describes the fields in the vmstat command output.

TABLE 13–1 Output From the vmstat Command

Category	Field Name	Description
procs		Reports on the following:
	r	The number of kernel threads in the dispatch queue
	b	The number of blocked kernel threads that are waiting for resources
	w	The number of swapped out LWPs that are waiting for processing resources to finish
memory		Reports on usage of real memory and virtual memory:
	swap	Available swap space
	free	Size of the free list
page		Reports on page faults and paging activity, in units per second:
	re	Pages reclaimed
	mf	Minor faults and major faults
	pi	Kbytes paged in
	po	Kbytes paged out
	fr	Kbytes freed

TABLE 13–1 Output From the vmstat Command *(Continued)*

Category	Field Name	Description
	de	Anticipated memory that is needed by recently swapped-in processes
	sr	Pages scanned by the page daemon not currently in use. If sr does not equal zero, the page daemon has been running.
disk		Reports the number of disk operations per second, showing data on up to four disks
faults		Reports the trap/interrupt rates per second:
	in	Interrupts per second
	sy	System calls per second
	cs	CPU context switch rate
cpu		Reports on the use of CPU time:
	us	User time
	sy	System time
	id	Idle time

For a more detailed description of this command, see the vmstat(1M) man page.

▼ How to Display Virtual Memory Statistics (vmstat)

- **Collect virtual memory statistics by using the vmstat command with a time interval in seconds.**

```
$ vmstat n
```

where *n* is the interval in seconds between reports.

Example 13–1 Displaying Virtual Memory Statistics

The following example shows the vmstat display of statistics that were gathered at five-second intervals.

```
$ vmstat 5
kthr      memory            page            disk          faults      cpu
 r b w   swap  free  re  mf pi po fr de sr dd f0 s1 --   in   sy   cs us sy id
 0 0 0 863160 365680  0   3  1  0  0  0  0  0  0  0  0  406  378  209  1  0 99
 0 0 0 765640 208568  0  36  0  0  0  0  0  0  0  0  0  479 4445 1378  3  3 94
 0 0 0 765640 208568  0   0  0  0  0  0  0  0  0  0  0  423  214  235  0  0 100
 0 0 0 765712 208640  0   0  0  0  0  0  0  3  0  0  0  412  158  181  0  0 100
 0 0 0 765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  157  179  0  0 100
 0 0 0 765832 208760  0   0  0  0  0  0  0  0  0  0  0  403  153  182  0  0 100
 0 0 0 765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  168  177  0  0 100
 0 0 0 765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  153  178  0  0 100
 0 0 0 765832 208760  0  18  0  0  0  0  0  0  0  0  0  407  165  186  0  0 100
```

▼ How to Display System Event Information (vmstat -s)

- **Run the** `vmstat -s` **command to show how many system events have taken place since the last time the system was booted.**

```
$ vmstat -s
        0 swap ins
        0 swap outs
        0 pages swapped in
        0 pages swapped out
   522586 total address trans. faults taken
    17006 page ins
       25 page outs
    23361 pages paged in
       28 pages paged out
    45594 total reclaims
    45592 reclaims from free list
        0 micro (hat) faults
   522586 minor (as) faults
    16189 major faults
    98241 copy-on-write faults
   137280 zero fill page faults
    45052 pages examined by the clock daemon
        0 revolutions of the clock hand
       26 pages freed by the clock daemon
     2857 forks
       78 vforks
     1647 execs
 34673885 cpu context switches
 65943468 device interrupts
   711250 traps
 63957605 system calls
  3523925 total name lookups (cache hits 99%)
    92590 user   cpu
    65952 system cpu
 16085832 idle   cpu
     7450 wait   cpu
```

How to Display Swapping Statistics (vmstat -S)

- **Run `vmstat -S` to show swapping statistics.**

```
$ vmstat -S
kthr      memory            page            disk          faults      cpu
 r b w   swap  free  si  so pi po fr de sr dd f0 s1 --   in   sy   cs us sy id
 0 0 0 862608 364792  0   0  1  0  0  0  0  0  0  0  0  406  394  213  1  0 99
```

The swapping statistics fields are described in the following list. For a description of the other fields, see Table 13–1.

`si` Average number of LWPs that are swapped in per second

`so` Number of whole processes that are swapped out

Note – The `vmstat` command truncates the output of `si` and `so` fields. Use the `sar` command to display a more accurate accounting of swap statistics.

How to Display Interrupts Per Device (vmstat -i)

- **Run the `vmstat -i` command to show the number of interrupts per device.**

Example 13–2 Displaying Interrupts Per Device

The following example shows output from the `vmstat -i` command.

```
$ vmstat -i
interrupt         total     rate
--------------------------------
clock          52163269      100
esp0            2600077        4
zsc0              25341        0
zsc1              48917        0
cgsixc0             459        0
lec0             400882        0
fdc0                 14        0
bppc0                 0        0
audiocs0              0        0
--------------------------------
Total          55238959      105
```

Displaying Disk Utilization Information (iostat)

Use the iostat command to report statistics about disk input and output, and to produce measures of throughput, utilization, queue lengths, transaction rates, and service time. For a detailed description of this command, refer to the iostat(1M) man page.

▼ How to Display Disk Utilization Information (iostat)

- **You can display disk utilization information by using the iostat command with a time interval in seconds.**

```
$ iostat 5
     tty          fd0           sd3          nfs1         nfs31          cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
  0    1   0   0  410    3   0   29    0   0    9    3   0   47    4  2  0 94
```

The first line of output shows the statistics since the last time the system was booted. Each subsequent line shows the interval statistics. The default is to show statistics for the terminal (tty), disks (fd and sd), and CPU (cpu).

Example 13–3 Displaying Disk Utilization Information

The following example shows disk statistics that were gathered every five seconds.

```
$ iostat 5
tty        sd0           sd6           nfs1         nfs49           cpu
tin tout kps tps serv  kps tps serv  kps tps serv  kps tps serv  us sy wt id
 0    0   1   0   49    0   0    0    0   0    0    0   0   15   0  0  0 100
 0   47   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16  44   6  132    0   0    0    0   0    0    0   0    0   0  0  1 99
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   3   1   23    0   0    0    0   0    0    0   0    0   0  0  1 99
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
 0   16   0   0    0    0   0    0    0   0    0    0   0    0   0  0  0 100
```

The following table describes the fields in the output of the iostat *n* command.

Device Type	Field Name	Description
Terminal	Device Type	
	`tin`	Number of characters in the terminal input queue
	`tout`	Number of characters in the terminal output queue
Disk	Device Type	
	`bps`	Blocks per second
	`tps`	Transactions per second
	`serv`	Average service time, in milliseconds
CPU	Device Type	
	`us`	In user mode
	`sy`	In system mode
	`wt`	Waiting for I/O
	`id`	Idle

▼ How to Display Extended Disk Statistics (`iostat -xtc`)

- **Run the `iostat -xtc` command to display extended disk statistics.**

```
$ iostat -xtc
                  extended device statistics                      tty         cpu
device       r/s    w/s   kr/s    kw/s wait actv  svc_t  %w  %b  tin tout  us sy wt id
fd0          0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0    0    0   0  0  0 100
sd0          0.0    0.0    0.4    0.4  0.0  0.0   49.5   0   0
sd6          0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs1         0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs49        0.0    0.0    0.0    0.0  0.0  0.0   15.1   0   0
nfs53        0.0    0.0    0.4    0.0  0.0  0.0   24.5   0   0
nfs54        0.0    0.0    0.0    0.0  0.0  0.0    6.3   0   0
nfs55        0.0    0.0    0.0    0.0  0.0  0.0    4.9   0   0
```

The `iostat -xtc` command displays a line of output for each disk. The output fields are described in the following list.

`r/s` Reads per second

`w/s` Writes per second

`kr/s` Kbytes read per second

`kw/s`	Kbytes written per second
`wait`	Average number of transactions that are waiting for service (queue length)
`actv`	Average number of transactions that are actively being serviced
`svc_t`	Average service time, in milliseconds
`%w`	Percentage of time that the queue is not empty
`%b`	Percentage of time that the disk is busy

Displaying Disk Space Statistics (`df`)

Use the `df` command to show the amount of free disk space on each mounted disk. The *usable* disk space that is reported by `df` reflects only 90 percent of full capacity, as the reporting statistics allows for 10 percent above the total available space. This *head room* normally stays empty for better performance.

The percentage of disk space actually reported by the `df` command is used space divided by usable space.

If the file system exceeds 90 percent capacity, you could transfer files to a disk that is not as full by using the `cp` command. Alternately, you could transfer files to a tape by using the `tar` or `cpio` commands. Or, you could remove the files.

For a detailed description of this command, see the `df`(1M) man page.

▼ How to Display Disk Space Information (`df -k`)

- **Use the `df -k` command to display disk space information in Kbytes.**

```
$ df -k
Filesystem           kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0    192807   40231  133296    24%    /
```

Example 13–4 Displaying File System Information

The following example shows the output from the `df -k` command.

```
$ df -k
Filesystem           kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0    254966  204319   25151    90%    /
/devices                  0       0       0     0%    /devices
ctfs                      0       0       0     0%    /system/contract
```

```
proc                       0       0       0     0%    /proc
mnttab                     0       0       0     0%    /etc/mnttab
swap                  496808     376  496432     1%    /etc/svc/volatile
objfs                      0       0       0     0%    /system/object
/dev/dsk/c0t0d0s6    3325302 3073415  218634    94%    /usr
fd                         0       0       0     0%    /dev/fd
swap                  496472      40  496432     1%    /var/run
swap                  496472      40  496432     1%    /tmp
/dev/dsk/c0t0d0s5      13702    1745   10587    15%    /opt
/dev/dsk/c0t0d0s7       9450    1045    7460    13%    /export/home
```

The following table describes the output of the `df -k` command.

Field Name	Description
`kbytes`	Total size of usable space in the file system
`used`	Amount of space used
`avail`	Amount of space available for use
`capacity`	Amount of space used, as a percentage of the total capacity
`mounted on`	Mount point

Monitoring System Activities (Task Map)

Task	Description	For Instructions
Check file access.	Display file access operation status by using the `sar` command with the `-a` option.	"How to Check File Access (`sar -a`)" on page 193
Check buffer activity.	Display buffer activity statistics by using the `sar` command with the `-b` option.	"How to Check Buffer Activity (`sar -b`)" on page 194
Check system call statistics.	Display system call statistics by using the `sar` command with the `-c` option.	"How to Check System Call Statistics (`sar -c`)" on page 195
Check disk activity.	Check disk activity by using the `sar` command with the `-d` option.	"How to Check Disk Activity (`sar -d`)" on page 197
Check page-out and memory.	Use the `sar` command with the `-g` option to display page-out memory freeing activities.	"How to Check Page-Out and Memory (`sar -g`)" on page 198

Task	Description	For Instructions
Check kernel memory allocation.	The kernel memory allocation (KMA) allows a kernel subsystem to allocate and free memory, as needed. Use the sar command with the -k option to check KMA.	"How to Check Kernel Memory Allocation (sar -k)" on page 200
Check interprocess communication.	Use the sar command with the -m option to report interprocess communication activities.	"How to Check Interprocess Communication (sar -m)" on page 202
Check page-in activity.	Use the sar command with the -p option to report page-in activity.	"How to Check Page-In Activity (sar -p)" on page 203
Check queue activity.	Use the sar command with the -q option to check the following: ■ Average queue length while queue is occupied ■ Percentage of time that the queue is occupied	"How to Check Queue Activity (sar -q)" on page 204
Check unused memory.	Use the sar command with the -r option to report the number of memory pages and swap file disk blocks that are currently used.	"How to Check Unused Memory (sar -r)" on page 205
Check CPU utilization.	Use the sar command with the -u option to display CPU utilization statistics.	"How to Check CPU Utilization (sar -u)" on page 206
Check system table status.	Use the sar command with the -v option to report status on the following system tables: ■ Process ■ Inode ■ File ■ Shared memory record	"How to Check System Table Status (sar -v)" on page 208
Check swapping activity.	Use the sar command with the -w option to check swapping activity.	"How to Check Swapping Activity (sar -w)" on page 209
Check terminal activity.	Use the sar command with the -y option to monitor terminal device activity.	"How to Check Terminal Activity (sar -y)" on page 210
Check overall system performance.	The sar -A command displays statistics from all options to provide overall system performance information.	"How to Check Overall System Performance (sar -A)" on page 211
Set up automatic data collection.	To set up your system to collect data automatically and to run the sar commands, do the following: ■ Run the svcadm enable system/sar:default command ■ Edit the /var/spool/cron/crontabs/sys file	"How to Set Up Automatic Data Collection" on page 214

Monitoring System Activities (sar)

Use the sar command to perform the following tasks:

- Organize and view data about system activity.
- Access system activity data on a special request basis.
- Generate automatic reports to measure and monitor system performance, as well as special request reports to pinpoint specific performance problems. For information on how to set up the sar command to run on your system, as well as a description of these tools, see "Collecting System Activity Data Automatically (sar)" on page 212.

For a detailed description of this command, see the sar(1) man page.

▼ How to Check File Access (sar -a)

● **Display file access operation statistics with the sar -a command.**

```
$ sar -a

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:00  iget/s namei/s dirbk/s
01:00:00       0       3       0
02:00:00       0       3       0
03:00:00       0       3       0
04:00:00       0       3       0
05:00:00       0       3       0
06:00:00       0       3       0
07:00:00       0       3       0
08:00:00       0       3       0
08:20:01       0       3       0
08:40:00       0       3       0
09:00:00       0       3       0
09:20:01       0      10       0
09:40:01       0       1       0
10:00:02       0       5       0

Average        0       4       0
```

The following list describes the field names and description of operating system routines that are reported by the sar -a command.

iget/s The number of requests made for inodes that were not in the directory name look-up cache (DNLC).

`namei/s` The number of file system path searches per second. If `namei` does not find a directory name in the DNLC, it calls `iget` to get the inode for either a file or directory. Hence, most `igets` are the result of DNLC misses.

`dirbk/s` The number of directory block reads issued per second.

The larger the reported values for these operating system routines, the more time the kernel is spending to access user files. The amount of time reflects how heavily programs and applications are using the file systems. The `-a` option is helpful for viewing how disk-dependent an application is.

▼ How to Check Buffer Activity (`sar -b`)

- **Display buffer activity statistics with the `sar -b` command.**

The buffer is used to cache metadata. Metadata includes inodes, cylinder group blocks, and indirect blocks.

```
$ sar -b
00:00:00 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00       0       0     100       0       0      55       0       0
```

Example 13–5 Checking Buffer Activity (`sar -b`)

The following example of `sar -b` command output shows that the `%rcache` and `%wcache` buffers are not causing any slowdowns. All the data is within acceptable limits.

```
$ sar -b

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00       0       0     100       0       0      94       0       0
02:00:01       0       0     100       0       0      94       0       0
03:00:00       0       0     100       0       0      92       0       0
04:00:00       0       1     100       0       1      94       0       0
05:00:00       0       0     100       0       0      93       0       0
06:00:00       0       0     100       0       0      93       0       0
07:00:00       0       0     100       0       0      93       0       0
08:00:00       0       0     100       0       0      93       0       0
08:20:00       0       1     100       0       1      94       0       0
08:40:01       0       1     100       0       1      93       0       0
09:00:00       0       1     100       0       1      93       0       0
09:20:00       0       1     100       0       1      93       0       0
09:40:00       0       2     100       0       1      89       0       0
10:00:00       0       9     100       0       5      92       0       0
```

```
10:20:00        0        0      100        0        0       68        0        0
10:40:00        0        1       98        0        1       70        0        0
11:00:00        0        1      100        0        1       75        0        0

Average         0        1      100        0        1       91        0        0
```

The following table describes the buffer activities that are displayed by the `-b` option.

Field Name	Description
`bread/s`	Average number of reads per second that are submitted to the buffer cache from the disk
`lread/s`	Average number of logical reads per second from the buffer cache
`%rcache`	Fraction of logical reads that are found in the buffer cache (100 % minus the ratio of `bread/s` to `lread/s`)
`bwrit/s`	Average number of physical blocks (512 blocks) that are written from the buffer cache to disk, per second
`lwrit/s`	Average number of logical writes to the buffer cache, per second
`%wcache`	Fraction of logical writes that are found in the buffer cache (100 % minus the ratio of `bwrit/s` to `lwrit/s`)
`pread/s`	Average number of physical reads, per second, that use character device interfaces
`pwrit/s`	Average number of physical write requests, per second, that use character device interfaces

The most important entries are the cache hit ratios `%rcache` and `%wcache`. These entries measure the effectiveness of system buffering. If `%rcache` falls below 90 percent, or if `%wcache` falls below 65 percent, it might be possible to improve performance by increasing the buffer space.

▼ How to Check System Call Statistics (`sar -c`)

- **Display system call statistics by using the `sar -c` command.**

```
$ sar -c
00:00:00 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      38       2       2    0.00    0.00     149     120
```

Example 13–6 Checking System Call Statistics (sar -c)

The following example shows output from the sar -c command.

```
$ sar -c

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      89      14       9    0.01    0.00    2906    2394
02:00:01      89      14       9    0.01    0.00    2905    2393
03:00:00      89      14       9    0.01    0.00    2908    2393
04:00:00      90      14       9    0.01    0.00    2912    2393
05:00:00      89      14       9    0.01    0.00    2905    2393
06:00:00      89      14       9    0.01    0.00    2905    2393
07:00:00      89      14       9    0.01    0.00    2905    2393
08:00:00      89      14       9    0.01    0.00    2906    2393
08:20:00      90      14       9    0.01    0.01    2914    2395
08:40:01      90      14       9    0.01    0.00    2914    2396
09:00:00      90      14       9    0.01    0.01    2915    2396
09:20:00      90      14       9    0.01    0.01    2915    2396
09:40:00     880     207     156    0.08    0.08   26671    9290
10:00:00    2020     530     322    0.14    0.13   57675   36393
10:20:00     853     129      75    0.02    0.01   10500    8594
10:40:00    2061     524     450    0.08    0.08  579217  567072
11:00:00    1658     404     350    0.07    0.06 1152916 1144203

Average      302      66      49    0.02    0.01   57842   55544
```

The following table describes the system call categories that are reported by the -c option. Typically, reads and writes account for about half of the total system calls. However, the percentage varies greatly with the activities that are being performed by the system.

Field Name	Description
scall/s	The number of all types of system calls per second, which is generally about 30 per second on a system with 4 to 6 users.
sread/s	The number of read system calls per second.
swrit/s	The number of write system calls per second.
fork/s	The number of fork system calls per second, which is about 0.5 per second on a system with 4 to 6 users. This number increases if shell scripts are running.

Field Name	Description
`exec/s`	The number of exec system calls per second. If `exec/s` divided by `fork/s` is greater than 3, look for inefficient `PATH` variables.
`rchar/s`	The number of characters (bytes) transferred by `read` system calls per second.
`wchar/s`	The number of characters (bytes) transferred by `write` system calls per second.

▼ How to Check Disk Activity (`sar -d`)

● **Display disk activity statistics with the `sar -d` command.**

```
$ sar -d

00:00:00   device        %busy   avque   r+w/s  blks/s  avwait  avserv
```

Example 13–7 Checking Disk Activity

This abbreviated example illustrates the output from the `sar -d` command.

```
$ sar -d

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

12:36:32   device        %busy   avque   r+w/s  blks/s  avwait  avserv

12:40:01   dad1             15     0.7      26     399    18.1    10.0
           dad1,a           15     0.7      26     398    18.1    10.0
           dad1,b            0     0.0       0       1     1.0     3.0
           dad1,c            0     0.0       0       0     0.0     0.0
           dad1,h            0     0.0       0       0     0.0     6.0
           fd0               0     0.0       0       0     0.0     0.0
           nfs1              0     0.0       0       0     0.0     0.0
           nfs2              1     0.0       1      12     0.0    13.2
           nfs3              0     0.0       0       2     0.0     1.9
           nfs4              0     0.0       0       0     0.0     7.0
           nfs5              0     0.0       0       0     0.0    57.1
           nfs6              1     0.0       6     125     4.3     3.2
           nfs7              0     0.0       0       0     0.0     6.0
           sd1               0     0.0       0       0     0.0     5.4
           ohci0,bu          0     0.0       0       0     0.0     0.0
           ohci0,ct          0     0.0       0       0     0.0     0.0
           ohci0,in          0     0.0       7       0     0.0     0.0
```

```
            ohci0,is           0     0.0       0       0     0.0     0.0
            ohci0,to           0     0.0       7       0     0.0     0.0
```

The following table describes the disk device activities that are reported by the -d option.

Field Name	Description
device	Name of the disk device that is being monitored.
%busy	Portion of time the device was busy servicing a transfer request.
avque	Average number of requests during the time the device was busy servicing a transfer request.
r+w/s	Number of read-and-write transfers to the device, per second.
blks/s	Number of 512-byte blocks that are transferred to the device, per second.
avwait	Average time, in milliseconds, that transfer requests wait idly in the queue. This time is measured only when the queue is occupied.
avserv	Average time, in milliseconds, for a transfer request to be completed by the device. For disks, this value includes seek times, rotational latency times, and data transfer times.

Note that queue lengths and wait times are measured when something is in the queue. If %busy is small, large queues and service times probably represent the periodic efforts by the system to ensure that altered blocks are promptly written to the disk.

▼ How to Check Page-Out and Memory (sar -g)

- **Use the sar -g command to display page-out and memory freeing activities in averages.**

```
$ sar -g
00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00     0.00     0.00     0.00     0.00     0.00
```

The output displayed by the sar -g command is a good indicator of whether more memory might be needed. Use the ps -elf command to show the number of cycles that are used by the page daemon. A high number of cycles, combined with high values for the pgfree/s and pgscan/s fields, indicates a memory shortage.

The sar -g command also shows whether inodes are being recycled too quickly and causing a loss of reusable pages.

Example 13–8 Checking Page-Out and Memory (sar -g)

The following example shows output from the sar -g command.

```
$ sar -g

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00     0.00     0.00     0.00     0.00     0.00
02:00:00     0.01     0.01     0.01     0.00     0.00
03:00:00     0.00     0.00     0.00     0.00     0.00
04:00:00     0.00     0.00     0.00     0.00     0.00
05:00:00     0.00     0.00     0.00     0.00     0.00
06:00:00     0.00     0.00     0.00     0.00     0.00
07:00:00     0.00     0.00     0.00     0.00     0.00
08:00:00     0.00     0.00     0.00     0.00     0.00
08:20:01     0.00     0.00     0.00     0.00     0.00
08:40:00     0.00     0.00     0.00     0.00     0.00
09:00:00     0.00     0.00     0.00     0.00     0.00
09:20:01     0.05     0.52     1.62    10.16     0.00
09:40:01     0.03     0.44     1.47     4.77     0.00
10:00:02     0.13     2.00     4.38    12.28     0.00
10:20:03     0.37     4.68    12.26    33.80     0.00

Average      0.02     0.25     0.64     1.97     0.00
```

The following table describes the output from the -g option.

Field Name	Description
pgout/s	The number of page-out requests per second.
ppgout/s	The actual number of pages that are paged-out, per second. A single page-out request might involve paging-out multiple pages.
pgfree/s	The number of pages, per second, that are placed on the free list.
pgscan/s	The number of pages, per second, that are scanned by the page daemon. If this value is high, the page daemon is spending a lot of time checking for free memory. This situation implies that more memory might be needed.

Field Name	Description
%ufs_ipf	The percentage of ufs inodes taken off the free list by iget that had reusable pages associated with them. These pages are flushed and cannot be reclaimed by processes. Thus, this field represents the percentage of igets with page flushes. A high value indicates that the free list of inodes is page-bound, and that the number of ufs inodes might need to be increased.

Checking Kernel Memory Allocation

The KMA allows a kernel subsystem to allocate and free memory, as needed.

Rather than statically allocating the maximum amount of memory it is expected to require under peak load, the KMA divides requests for memory into three categories:

- Small (less than 256 bytes)
- Large (512 bytes to 4 Kbytes)
- Oversized (greater than 4 Kbytes)

The KMA keeps two pools of memory to satisfy small requests and large requests. The oversized requests are satisfied by allocating memory from the system page allocator.

If you are checking a system that is being used to write drivers or STREAMS that use KMA resources, then the sar -k command will likely prove useful. Otherwise, you will probably not need the information it provides. Any driver or module that uses KMA resources, but does not specifically return the resources before it exits, can create a memory leak. A memory leak causes the amount of memory that is allocated by KMA to increase over time. Thus, if the alloc fields of the sar -k command increase steadily over time, there might be a memory leak. Another indication of a memory leak is failed requests. If this problem occurs, a memory leak has probably caused KMA to be unable to reserve and allocate memory.

If it appears that a memory leak has occurred, you should check any drivers or STREAMS that might have requested memory from KMA and not returned it.

▼ How to Check Kernel Memory Allocation (sar -k)

- **Use the sar -k command to report on the following activities of the Kernel Memory Allocator (KMA).**

```
$ sar -k
00:00:00 sml_mem   alloc  fail  lg_mem   alloc   fail  ovsz_alloc  fail
01:00:00 2523136 1866512     0 18939904 14762364    0      360448     0
02:00:02 2523136 1861724     0 18939904 14778748    0      360448     0
```

Example 13–9 Checking Kernel Memory Allocation (sar -k)

The following is an abbreviated example of sar -k output.

```
$ sar -k

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 sml_mem   alloc  fail  lg_mem   alloc  fail  ovsz_alloc  fail
01:00:00 6119744 4852865     0 60243968 54334808   156     9666560     0
02:00:01 6119744 4853057     0 60243968 54336088   156     9666560     0
03:00:00 6119744 4853297     0 60243968 54335760   156     9666560     0
04:00:00 6119744 4857673     0 60252160 54375280   156     9666560     0
05:00:00 6119744 4858097     0 60252160 54376240   156     9666560     0
06:00:00 6119744 4858289     0 60252160 54375608   156     9666560     0
07:00:00 6119744 4858793     0 60252160 54442424   156     9666560     0
08:00:00 6119744 4858985     0 60252160 54474552   156     9666560     0
08:20:00 6119744 4858169     0 60252160 54377400   156     9666560     0
08:40:01 6119744 4857345     0 60252160 54376880   156     9666560     0
09:00:00 6119744 4859433     0 60252160 54539752   156     9666560     0
09:20:00 6119744 4858633     0 60252160 54410920   156     9666560     0
09:40:00 6127936 5262064     0 60530688 55619816   156     9666560     0
10:00:00 6545728 5823137     0 62996480 58391136   156     9666560     0
10:20:00 6545728 5758997     0 62996480 57907400   156     9666560     0
10:40:00 6734144 6035759     0 64389120 59743064   156    10493952     0
11:00:00 6996288 6394872     0 65437696 60935936   156    10493952     0

Average  6258044 5150556     0 61138340 55609004   156     9763900     0
```

The following table describes the output from the -k option.

Field Name	Description
sml_mem	The amount of memory, in bytes, that the KMA has available in the small memory request pool. In this pool, here a small request is less than 256 bytes.
alloc	The amount of memory, in bytes, that the KMA has allocated from its small memory request pool to small memory requests.
fail	The number of requests for small amounts of memory that failed.
lg_mem	The amount of memory, in bytes, that the KMA has available in the large memory request pool. In this pool, a large request is from 512 bytes to 4 Kbytes.

Field Name	Description
alloc	The amount of memory, in bytes, that the KMA has allocated from its large memory request pool to large memory requests.
fail	The number of failed requests for large amounts of memory.
ovsz_alloc	The amount of memory that is allocated for oversized requests, which are requests that are greater than 4 Kbytes. These requests are satisfied by the page allocator. Thus, there is no pool.
fail	The number of failed requests for oversized amounts of memory.

▼ How to Check Interprocess Communication (sar -m)

- **Use the sar -m command to report interprocess communication activities.**

```
$ sar -m
00:00:00   msg/s  sema/s
01:00:00    0.00    0.00
```

These figures are usually zero (0.00), unless you are running applications that use messages or semaphores.

The following list describes the output from the -m option.

msg/s The number of message operations (sends and receives) per second

sema/s The number of semaphore operations per second

Example 13–10 Checking Interprocess Communication (sar -m)

The following abbreviated example shows output from the sar -m command.

```
$ sar -m

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:00   msg/s  sema/s
01:00:00    0.00    0.00
02:00:02    0.00    0.00
03:00:00    0.00    0.00
04:00:00    0.00    0.00
05:00:01    0.00    0.00
06:00:00    0.00    0.00
```

```
Average     0.00    0.00
```

▼ How to Check Page-In Activity (sar -p)

- **Use the `sar -p` command to report page-in activity, which includes protection and translation faults.**

```
$ sar -p
00:00:00  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
01:00:00    0.07    0.00    0.00    0.21    0.39    0.00
```

Example 13–11 Checking Page-In Activity (sar -p)

The following example shows output from the `sar -p` command.

```
$ sar -p

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04  atch/s  pgin/s ppgin/s  pflt/s  vflt/s slock/s
01:00:00    0.09    0.00    0.00    0.78    2.02    0.00
02:00:01    0.08    0.00    0.00    0.78    2.02    0.00
03:00:00    0.09    0.00    0.00    0.81    2.07    0.00
04:00:00    0.11    0.01    0.01    0.86    2.18    0.00
05:00:00    0.08    0.00    0.00    0.78    2.02    0.00
06:00:00    0.09    0.00    0.00    0.78    2.02    0.00
07:00:00    0.08    0.00    0.00    0.78    2.02    0.00
08:00:00    0.09    0.00    0.00    0.78    2.02    0.00
08:20:00    0.11    0.00    0.00    0.87    2.24    0.00
08:40:01    0.13    0.00    0.00    0.90    2.29    0.00
09:00:00    0.11    0.00    0.00    0.88    2.24    0.00
09:20:00    0.10    0.00    0.00    0.88    2.24    0.00
09:40:00    2.91    1.80    2.38    4.61   17.62    0.00
10:00:00    2.74    2.03    3.08    8.17   21.76    0.00
10:20:00    0.16    0.04    0.04    1.92    2.96    0.00
10:40:00    2.10    2.50    3.42    6.62   16.51    0.00
11:00:00    3.36    0.87    1.35    3.92   15.12    0.00

Average     0.42    0.22    0.31    1.45    4.00    0.00
```

The following table describes the reported statistics from the `-p` option.

Field Name	Description
`atch/s`	The number of page faults, per second, that are satisfied by reclaiming a page currently in memory (attaches per second). Instances include reclaiming an invalid page from the free list and sharing a page of text that is currently being used by another process. An example is two or more processes that are accessing the same program text.
`pgin/s`	The number of times, per second, that file systems receive page-in requests.
`ppgin/s`	The number of pages paged in, per second. A single page-in request, such as a soft-lock request (see `slock/s`) or a large block size, might involve paging-in multiple pages.
`pflt/s`	The number of page faults from protection errors. Instances of protection faults indicate illegal access to a page and "copy-on-writes." Generally, this number consists primarily of "copy-on-writes."
`vflt/s`	The number of address translation page faults, per second. These faults are known as validity faults. Validity faults occur when a valid process table entry does not exist for a given virtual address.
`slock/s`	The number of faults, per second, caused by software lock requests that require physical I/O. An example of the occurrence of a soft-lock request is the transfer of data from a disk to memory. The system locks the page that is to receive the data so that the page cannot be claimed and used by another process.

▼ How to Check Queue Activity (`sar -q`)

● **Use the `sar -q` command to report the following information:**

- The Average queue length while the queue is occupied.
- The percentage of time that the queue is occupied.

```
$ sar -q
00:00:00 runq-sz %runocc swpq-sz %swpocc
```

The following list describes the output from the `-q` option.

`runq-sz` The number of kernel threads in memory that are waiting for a CPU to run. Typically, this value should be less than 2. Consistently higher values mean that the system might be CPU-bound.

%runocc The percentage of time that the dispatch queues are occupied.

swpq-sz No longer reported by the sar command.

%swpocc No longer reported by the sar command.

Example 13–12 Checking Queue Activity

The following example shows output from the sar -q command. If the %runocc value is high (greater than 90 percent) and the runq-sz value is greater than 2, the CPU is heavily loaded and response is degraded. In this case, additional CPU capacity might be required to obtain acceptable system response.

```
$ sar -q

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 runq-sz %runocc swpq-sz %swpocc
01:00:00     1.0       0     0.0       0
02:00:01     1.3       0     0.0       0
03:00:00     1.0       0     0.0       0
04:00:00     1.0       0     0.0       0
05:00:00     1.0       0     0.0       0
06:00:00     2.0       0     0.0       0
07:00:00     0.0       0     0.0       0
08:00:00     1.0       0     0.0       0
08:20:00     1.0       0     0.0       0
08:40:01     2.0       0     0.0       0
09:00:00     0.0       0     0.0       0
09:20:00     1.0       0     0.0       0
09:40:00     1.2       2     0.0       0
10:00:00     1.2       2     0.0       0
10:20:00     1.0       1     0.0       0
10:40:00     1.3       9     0.0       0
11:00:00     1.2       7     0.0       0

Average      1.2       1     0.0       0
```

▼ How to Check Unused Memory (sar -r)

- **Use the sar -r command to report the number of memory pages and swap-file disk blocks that are currently unused.**

```
$ sar -r
00:00:00 freemem freeswap
01:00:00    2135   401922
```

The following list describes the output from the -r option.

freemem The average number of memory pages that are available to user processes over the intervals sampled by the command. Page size is machine-dependent.

freeswap The number of 512-byte disk blocks that are available for page swapping.

Example 13–13 Checking Unused Memory (sar -r)

The following example shows output from the sar -r command.

```
$ sar -r

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 freemem freeswap
01:00:00   44717  1715062
02:00:01   44733  1715496
03:00:00   44715  1714746
04:00:00   44751  1715403
05:00:00   44784  1714743
06:00:00   44794  1715186
07:00:00   44793  1715159
08:00:00   44786  1714914
08:20:00   44805  1715576
08:40:01   44797  1715347
09:00:00   44761  1713948
09:20:00   44802  1715478
09:40:00   41770  1682239
10:00:00   35401  1610833
10:20:00   34295  1599141
10:40:00   33943  1598425
11:00:00   30500  1561959

Average    43312  1699242
```

▼ How to Check CPU Utilization (sar -u)

- **Use the sar -u command to display CPU utilization statistics.**

```
$ sar -u
00:00:00    %usr    %sys    %wio   %idle
01:00:00       0       0       0     100
```

The sar command without any options is equivalent to the sar -u command. At any given moment, the processor is either busy or idle. When busy, the processor is in either user mode or system mode. When idle, the processor is either waiting for I/O completion or “sitting still” with no work to do.

The following list describes output from the -u option.

%usr Lists the percentage of time that the processor is in user mode

%sys Lists the percentage of time that the processor is in system mode

%wio Lists the percentage of time that the processor is idle and waiting for I/O completion

%idle Lists the percentage of time that the processor is idle and not waiting for I/O

A high %wio value generally means that a disk slowdown has occurred.

Example 13–14 Checking CPU Utilization (sar -u)

The following example shows output from the sar -u command.

```
$ sar -u

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04    %usr    %sys    %wio   %idle
01:00:00       0       0       0     100
02:00:01       0       0       0     100
03:00:00       0       0       0     100
04:00:00       0       0       0     100
05:00:00       0       0       0     100
06:00:00       0       0       0     100
07:00:00       0       0       0     100
08:00:00       0       0       0     100
08:20:00       0       0       0      99
08:40:01       0       0       0      99
09:00:00       0       0       0      99
09:20:00       0       0       0      99
09:40:00       4       1       0      95
10:00:00       4       2       0      94
10:20:00       1       1       0      98
10:40:00      18       3       0      79
11:00:00      25       3       0      72

Average        2       0       0      98
```

How to Check System Table Status (sar -v)

- **Use the sar -v command to report the status of the process table, inode table, file table, and shared memory record table.**

```
$ sar -v
00:00:00  proc-sz    ov  inod-sz    ov  file-sz    ov   lock-sz
01:00:00   43/922     0 2984/4236    0  322/322     0    0/0
```

Example 13–15 Checking System Table Status (sar -v)

The following abbreviated example shows output from the sar -v command. This example shows that all tables are large enough to have no overflows. These tables are all dynamically allocated based on the amount of physical memory.

```
$ sar -v

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04  proc-sz    ov  inod-sz    ov  file-sz    ov   lock-sz
01:00:00   69/8010    0 3476/34703    0    0/0       0    0/0
02:00:01   69/8010    0 3476/34703    0    0/0       0    0/0
03:00:00   69/8010    0 3476/34703    0    0/0       0    0/0
04:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
05:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
06:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
07:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
08:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
08:20:00   69/8010    0 3494/34703    0    0/0       0    0/0
08:40:01   69/8010    0 3494/34703    0    0/0       0    0/0
09:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
09:20:00   69/8010    0 3494/34703    0    0/0       0    0/0
09:40:00   74/8010    0 3494/34703    0    0/0       0    0/0
10:00:00   75/8010    0 4918/34703    0    0/0       0    0/0
10:20:00   72/8010    0 4918/34703    0    0/0       0    0/0
10:40:00   71/8010    0 5018/34703    0    0/0       0    0/0
11:00:00   77/8010    0 5018/34703    0    0/0       0    0/0
```

Output from the -v option is described in the following table.

Field Name	Description
proc-sz	The number of process entries (proc structures) that are currently being used, or allocated, in the kernel.

Field Name	Description
inod-sz	The total number of inodes in memory compared to the maximum number of inodes that are allocated in the kernel. This number is not a strict high watermark. The number can overflow.
file-sz	The size of the open system file table. The sz is given as 0, because space is allocated dynamically for the file table.
ov	The overflows that occur between sampling points for each table.
lock-sz	The number of shared memory record table entries that are currently being used, or allocated, in the kernel. The sz is given as 0 because space is allocated dynamically for the shared memory record table.

▼ How to Check Swapping Activity (sar -w)

- **Use the sar -w command to report swapping and switching activity.**

```
$ sar -w
00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00    0.00     0.0    0.00     0.0      22
```

The following list describes target values and observations related to the sar -w command output.

swpin/s The number of LWP transfers into memory per second.

bswin/s The number of blocks transferred for swap-ins per second. /* (float)PGTOBLK(xx->cvmi.pgswapin) / sec_diff */

swpot/s The average number of processes that are swapped out of memory per second. If the number is greater than 1, you might need to increase memory.

bswot/s The number of blocks that are transferred for swap-outs per second.

pswch/s The number of kernel thread switches, per second.

Note – All process swap-ins include process initialization.

Example 13–16 Checking Swap Activity (sar -w)

The following example shows output from the sar -w command.

```
$ sar -w

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00    0.00     0.0    0.00     0.0     132
02:00:01    0.00     0.0    0.00     0.0     133
03:00:00    0.00     0.0    0.00     0.0     133
04:00:00    0.00     0.0    0.00     0.0     134
05:00:00    0.00     0.0    0.00     0.0     133
06:00:00    0.00     0.0    0.00     0.0     133
07:00:00    0.00     0.0    0.00     0.0     132
08:00:00    0.00     0.0    0.00     0.0     131
08:20:00    0.00     0.0    0.00     0.0     133
08:40:01    0.00     0.0    0.00     0.0     132
09:00:00    0.00     0.0    0.00     0.0     132
09:20:00    0.00     0.0    0.00     0.0     132
09:40:00    0.00     0.0    0.00     0.0     335
10:00:00    0.00     0.0    0.00     0.0     601
10:20:00    0.00     0.0    0.00     0.0     353
10:40:00    0.00     0.0    0.00     0.0     747
11:00:00    0.00     0.0    0.00     0.0     804

Average     0.00     0.0    0.00     0.0     198
```

How to Check Terminal Activity (`sar -y`)

- **Use the `sar -y` command to monitor terminal device activities.**

```
$ sar -y
00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00       0       0       0       0       0       0
```

If you have a lot of terminal I/O, you can use this report to determine if any bad lines exist. The activities recorded are defined in the following list.

`rawch/s`	Input characters (raw queue) per second
`canch/s`	Input characters that are processed by canon (canonical queue) per second
`outch/s`	Output characters (output queue) per second
`rcvin/s`	Receiver hardware interrupts per second
`xmtin/s`	Transmitter hardware interrupts per second
`mdmin/s`	Modem interrupts per second

The number of modem interrupts per second (mdmin/s) should be close to zero. The receive and transmit interrupts per second (xmtin/s and rcvin/s) should be less than or equal to the number of incoming or outgoing characters, respectively. If not, check for bad lines.

Example 13–17 Checking Terminal Activity (sar -y)

The following example shows output from the sar -y command.

```
$ sar -y

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00       0       0       0       0       0       0
02:00:01       0       0       0       0       0       0
03:00:00       0       0       0       0       0       0
04:00:00       0       0       0       0       0       0
05:00:00       0       0       0       0       0       0
06:00:00       0       0       0       0       0       0
07:00:00       0       0       0       0       0       0
08:00:00       0       0       0       0       0       0
08:20:00       0       0       0       0       0       0
08:40:01       0       0       0       0       0       0
09:00:00       0       0       0       0       0       0
09:20:00       0       0       0       0       0       0
09:40:00       0       0       1       0       0       0
10:00:00       0       0      37       0       0       0
10:20:00       0       0       0       0       0       0
10:40:00       0       0       3       0       0       0
11:00:00       0       0       3       0       0       0

Average        0       0       1       0       0       0
```

▼ How to Check Overall System Performance (sar -A)

- **Use the sar -A command to display statistics from all options to provide a view of overall system performance.**

  This command provides a more global perspective. If data from more than a single time segment is shown, the report includes averages.

Collecting System Activity Data Automatically (`sar`)

Three commands are involved in the automatic collection of system activity data: `sadc`, `sa1`, and `sa2`.

The `sadc` data collection utility periodically collects data on system activity and saves the data in a file in binary format, one file for each 24-hour period. You can set up the `sadc` command to run periodically (usually once each hour), and whenever the system boots to multiuser mode. The data files are placed in the `/var/adm/sa` directory. Each file is named `sa`*dd*, where *dd* is the current date. The format of the command is as follows:

```
/usr/lib/sa/sadc [t n] [ofile]
```

The command samples *n* times with an interval of *t* seconds, which should be greater than five seconds between samples. This command then writes to the binary *ofile* file, or to standard output.

Running the `sadc` Command When Booting

The `sadc` command should be run at system boot time to record the statistics from when the counters are reset to zero. To make sure that the `sadc` command is run at boot time, the `svcadm enable system/sar:default` command writes a record to the daily data file.

The command entry has the following format:

```
/usr/bin/su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa‘date +%d‘"
```

Running the `sadc` Command Periodically With the `sa1` Script

To generate periodic records, you need to run the `sadc` command regularly. The simplest way to do so is to uncomment the following lines in the `/var/spool/cron/crontabs/sys` file:

```
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

The sys crontab entries do the following:

- The first two crontab entries cause a record to be written to the /var/adm/sa/sa*dd* file every 20 minutes from 8 a.m. to 5 p.m., Monday through Friday, and every hour on the hour otherwise.
- The third entry writes a record to the /var/adm/sa/sar*dd* file hourly, Monday through Friday, and includes all sar options.

You can change these defaults to meet your needs.

Producing Reports With the sa2 Shell Script

Another shell script, sa2, produces reports rather than binary data files. The sa2 command invokes the sar command and writes the ASCII output to a report file.

Setting Up Automatic Data Collection (sar)

The sar command can be used either to gather system activity data itself or to report what has been collected in the daily activity files that are created by the sadc command.

The sar command has the following formats:

```
sar [-aAbcdgkmpqruvwy] [-o file] t [n]
```

```
sar [-aAbcdgkmpqruvwy] [-s time] [-e time] [-i sec] [-f file]
```

The following sar command samples cumulative activity counters in the operating system every *t* seconds, *n* times. The *t* should be five seconds or greater. Otherwise, the command itself might affect the sample. You must specify a time interval in which to take the samples. Otherwise, the command operates according to the second format. The default value of *n* is 1. The following example takes two samples separated by 10 seconds. If the -o option were specified, samples are saved in binary format.

```
$ sar -u 10 2
```

Other important information about the sar command includes the following:

- With no sampling interval or number of samples specified, the sar command extracts data from a previously recorded file. This file is either the file specified by the -f option or, by default, the standard daily activity file, /var/adm/sa/sa*dd*, for the most recent day.
- The -s and -e options define the starting time and the ending time for the report. Starting and ending times are of the form *hh*[:*mm*[:*ss*]], where *hh*, *mm*, and *ss* represent hours, minutes, and seconds.

- The -i option specifies, in seconds, the intervals between record selection. If the -i option is not included, all intervals that are found in the daily activity file are reported.

The following table lists the sar options and their actions.

TABLE 13–2 Options for the sar Command

Option	Actions
-a	Checks file access operations
-b	Checks buffer activity
-c	Checks system calls
-d	Checks activity for each block device
-g	Checks page-out and memory freeing
-k	Checks kernel memory allocation
-m	Checks interprocess communication
-nv	Checks system table status
-p	Checks swap and dispatch activity
-q	Checks queue activity
-r	Checks unused memory
-u	Checks CPU utilization
-w	Checks swapping and switching volume
-y	Checks terminal activity
-A	Reports overall system performance, which is the same as entering all options.

Using no option is equivalent to calling the sar command with the -u option.

▼ How to Set Up Automatic Data Collection

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Run the `svcadm enable system/sar:default` command.**

   This version of the sadc command writes a special record that marks the time when the counters are reset to zero (boot time).

3 **Edit the `/var/spool/cron/crontabs/sys` `crontab` file.**

Note – Do not edit a `crontab` file directly. Instead, use the `crontab -e` command to make changes to an existing `crontab` file.

```
# crontab -e sys
```

4 **Uncomment the following lines:**

```
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

For more information, see the `crontab(1)` man page.

CHAPTER 14

Troubleshooting Software Problems (Overview)

This chapter provides a general overview of troubleshooting software problems, including information on troubleshooting system crashes and viewing system messages.

This is a list of information in this chapter.

- "What's New in Troubleshooting?" on page 217
- "Where to Find Software Troubleshooting Tasks" on page 219
- "Troubleshooting a System Crash" on page 220
- "Troubleshooting a System Crash Checklist" on page 221

What's New in Troubleshooting?

This section describes new or changed troubleshooting information in this Solaris release.

For information on new or changed troubleshooting features in the Solaris 10 release, see the following:

- "Dynamic Tracing Facility" on page 218
- "kmdb Replaces kadb as Standard Solaris Kernel Debugger" on page 218

For a complete listing of new Solaris features and a description of Solaris releases, see *Solaris 10 What's New*.

Common Agent Container Problems

Solaris 10 6/06: The common agent container is a stand-alone Java program that is now included in the Solaris OS. This program implements a container for Java management applications. The common agent container provides a management infrastructure that is designed for Java Management Extensions (JMX) and Java Dynamic Management Kit (Java DMK) based functionality. The software is installed by the SUNWcacaort package and resides in the /usr/lib/cacao directory.

Typically, the container is not visible. However, there are two instances when you might need to interact with the container daemon:

- It is possible that another application might attempt to use a network port that is reserved for the common agent container.
- In the event that a certificate store is compromised, you might have to regenerate the common agent container certificate keys.

For information about how to troubleshoot these problems, see "Troubleshooting Common Agent Container Problems in the Solaris OS" on page 260.

x86: SMF Boot Archive Service Might Fail During System Reboot

Solaris 10 1/06: If a system crash occurs in the GRUB based boot environment, it is possible that the SMF service `svc:/system/boot-archive:default` might fail when the system is rebooted. If this problem occurs, reboot the system and select the Solaris failsafe archive in the GRUB boot menu. Follow the prompts to rebuild the boot archive. After the archive is rebuilt, reboot the system. To continue the boot process, you can use the `svcadm` command to clear the `svc:/system/boot-archive:default` service. For instructions, see "x86: What to Do if the SMF Boot Archive Service Fails During a System Reboot" on page 255. For more information on GRUB based booting, see "Booting an x86 Based System by Using GRUB (Task Map)" in *System Administration Guide: Basic Administration*.

Dynamic Tracing Facility

The Solaris Dynamic Tracing (DTrace) facility is a comprehensive dynamic tracking facility that gives you a new level of observerability into the Solaris kernel and user processes. DTrace helps you understand your system by permitting you to dynamically instrument the OS kernel and user processes to record data that you specify at locations of interest, called, *probes*. Each probe can be associated with custom programs that are written in the new D programming language. All of DTrace's instrumentation is entirely dynamic and available for use on your production system. For more information, see the `dtrace`(1M) man page and the *Solaris Dynamic Tracing Guide*.

`kmdb` Replaces `kadb` as Standard Solaris Kernel Debugger

`kmdb` has replaced `kadb` as the standard "in situ" Solaris kernel debugger.

kmdb brings all the power and flexibility of mdb to live kernel debugging. kmdb supports the following:

- Debugger commands (dcmds)
- Debugger modules (dmods)
- Access to kernel type data
- Kernel execution control
- Inspection
- Modification

For more information, see the kmdb(1) man page. For step-by-step instructions on using kmdb to troubleshoot a system, see "How to Boot the System With the Kernel Debugger (kmdb)" in *System Administration Guide: Basic Administration* and "How to Boot a System With the Kernel Debugger in the GRUB Boot Environment (kmdb)" in *System Administration Guide: Basic Administration*.

Where to Find Software Troubleshooting Tasks

Troubleshooting Task	For More Information
Manage system crash information	Chapter 17, "Managing System Crash Information (Tasks)"
Manage core files	Chapter 16, "Managing Core Files (Tasks)"
Troubleshoot software problems such as reboot failures and backup problems	Chapter 18, "Troubleshooting Miscellaneous Software Problems (Tasks)"
Troubleshoot file access problems	Chapter 19, "Troubleshooting File Access Problems (Tasks)"
Troubleshoot printing problems	Chapter 11, "Troubleshooting Printing Problems in the Solaris OS (Tasks)," in *System Administration Guide: Solaris Printing*
Resolve UFS file system inconsistencies	Chapter 20, "Resolving UFS File System Inconsistencies (Tasks)"
Troubleshoot software package problems	Chapter 21, "Troubleshooting Software Package Problems (Tasks)"

Additional Resources for Troubleshooting System and Software Problems

You can use the Sun Explorer software to collect data for troubleshooting system and software problems. For more information about downloading the Sun Explorer software, see *Sun Explorer User's Guide*.

Troubleshooting a System Crash

If a system running the Solaris Operating System crashes, provide your service provider with as much information as possible, including crash dump files.

What to Do if the System Crashes

The most important things to remember are:

1. Write down the system console messages.

   If a system crashes, making it run again might seem like your most pressing concern. However, before you reboot the system, examine the console screen for messages. These messages can provide some insight about what caused the crash. Even if the system reboots automatically and the console messages have disappeared from the screen, you might be able to check these messages by viewing the system error log, the`/var/adm/messages` file. For more information about viewing system error log files, see "How to View System Messages" on page 224.

   If you have frequent crashes and can't determine their cause, gather all the information you can from the system console or the `/var/adm/messages` files, and have it ready for a customer service representative to examine. For a complete list of troubleshooting information to gather for your service provider, see "Troubleshooting a System Crash" on page 220.

   If the system fails to reboot successfully after a system crash, see Chapter 18, "Troubleshooting Miscellaneous Software Problems (Tasks)."
2. Synchronize the disks and reboot.

   ```
   ok sync
   ```

   If the system fails to reboot successfully after a system crash, see Chapter 18, "Troubleshooting Miscellaneous Software Problems (Tasks)."

Check to see if a system crash dump was generated after the system crash. System crash dumps are saved by default. For information about crash dumps, see Chapter 17, "Managing System Crash Information (Tasks)."

Gathering Troubleshooting Data

Answer the following questions to help isolate the system problem. Use "Troubleshooting a System Crash Checklist" on page 221 for gathering troubleshooting data for a crashed system.

TABLE 14–1 Identifying System Crash Data

Question	Description
Can you reproduce the problem?	This is important because a reproducible test case is often essential for debugging really hard problems. By reproducing the problem, the service provider can build kernels with special instrumentation to trigger, diagnose, and fix the bug.
Are you using any third-party drivers?	Drivers run in the same address space as the kernel, with all the same privileges, so they can cause system crashes if they have bugs.
What was the system doing just before it crashed?	If the system was doing anything unusual like running a new stress test or experiencing higher-than-usual load, that might have led to the crash.
Were there any unusual console messages right before the crash?	Sometimes the system will show signs of distress before it actually crashes; this information is often useful.
Did you add any tuning parameters to the `/etc/system` *file*?	Sometimes tuning parameters, such as increasing shared memory segments so that the system tries to allocate more than it has, can cause the system to crash.
Did the problem start recently?	If so, did the onset of problems coincide with any changes to the system, for example, new drivers, new software, different workload, CPU upgrade, or a memory upgrade.

Troubleshooting a System Crash Checklist

Use this checklist when gathering system data for a crashed system.

Item	Your Data
Is a system crash dump available?	
Identify the operating system release and appropriate software application release levels.	
Identify system hardware.	
Include `prtdiag` output for sun4u systems. Include Explorer output for other systems.	

Item	Your Data
Are patches installed? If so, include `showrev -p` output.	
Is the problem reproducible?	
Does the system have any third-party drivers?	
What was the system doing before it crashed?	
Were there any unusual console messages right before the system crashed?	
Did you add any parameters to the `/etc/system` file?	
Did the problem start recently?	

◆ ◆ ◆ CHAPTER 15

Managing System Messages

This chapter describes system messaging features in the Solaris Operating System.

Viewing System Messages

System messages display on the console device. The text of most system messages look like this:

[ID *msgid facility*.*priority*]

For example:

```
[ID 672855 kern.notice] syncing file systems...
```

If the message originated in the kernel, the kernel module name is displayed. For example:

```
Oct 1 14:07:24 mars ufs: [ID 845546 kern.notice] alloc: /: file system full
```

When a system crashes, it might display a message on the system console like this:

```
panic: error message
```

Less frequently, this message might be displayed instead of the panic message:

```
Watchdog reset !
```

The error logging daemon, `syslogd`, automatically records various system warnings and errors in message files. By default, many of these system messages are displayed on the system console and are stored in the `/var/adm` directory. You can direct where these messages are stored by setting up system message logging. For more information, see "Customizing System Message Logging" on page 226. These messages can alert you to system problems, such as a device that is about to fail.

The /var/adm directory contains several message files. The most recent messages are in /var/adm/messages file (and in messages.*), and the oldest are in the messages.3 file. After a period of time (usually every ten days), a new messages file is created. The messages.0 file is renamed messages.1, messages.1 is renamed messages.2, and messages.2 is renamed messages.3. The current /var/adm/messages.3 file is deleted.

Because the /var/adm directory stores large files containing messages, crash dumps, and other data, this directory can consume lots of disk space. To keep the /var/adm directory from growing too large, and to ensure that future crash dumps can be saved, you should remove unneeded files periodically. You can automate this task by using the crontab file. For more information on automating this task, see "How to Delete Crash Dump Files" on page 91 and Chapter 8, "Scheduling System Tasks (Tasks)."

▼ How to View System Messages

- **Display recent messages generated by a system crash or reboot by using the dmesg command.**

```
$ dmesg
```

Or, use the more command to display one screen of messages at a time.

```
$ more /var/adm/messages
```

Example 15–1 Viewing System Messages

The following example shows output from the dmesg command.

```
$ dmesg
Jan  3 08:44:41 starbug genunix: [ID 540533 kern.notice] SunOS Release 5.10 ...
Jan  3 08:44:41 starbug genunix: [ID 913631 kern.notice] Copyright 1983-2003 ...
Jan  3 08:44:41 starbug genunix: [ID 678236 kern.info] Ethernet address ...
Jan  3 08:44:41 starbug unix: [ID 389951 kern.info] mem = 131072K (0x8000000)
Jan  3 08:44:41 starbug unix: [ID 930857 kern.info] avail mem = 121888768
Jan  3 08:44:41 starbug rootnex: [ID 466748 kern.info] root nexus = Sun Ultra 5/
10 UPA/PCI (UltraSPARC-IIi 333MHz)
Jan  3 08:44:41 starbug rootnex: [ID 349649 kern.info] pcipsy0 at root: UPA 0x1f0x0
Jan  3 08:44:41 starbug genunix: [ID 936769 kern.info] pcipsy0 is /pci@1f,0
Jan  3 08:44:41 starbug pcipsy: [ID 370704 kern.info] PCI-device: pci@1,1, simba0
Jan  3 08:44:41 starbug genunix: [ID 936769 kern.info] simba0 is /pci@1f,0/pci@1,1
Jan  3 08:44:41 starbug pcipsy: [ID 370704 kern.info] PCI-device: pci@1, simba1
Jan  3 08:44:41 starbug genunix: [ID 936769 kern.info] simba1 is /pci@1f,0/pci@1
Jan  3 08:44:57 starbug simba: [ID 370704 kern.info] PCI-device: ide@3, uata0
Jan  3 08:44:57 starbug genunix: [ID 936769 kern.info] uata0 is /pci@1f,0/pci@1,
1/ide@3
Jan  3 08:44:57 starbug uata: [ID 114370 kern.info] dad0 at pci1095,6460
.
```

```
.
.
```

See Also For more information, see the dmesg(1M) man page.

System Log Rotation

System log files are rotated by the `logadm` command from an entry in the root `crontab` file. The `/usr/lib/newsyslog` script is no longer used.

The system log rotation is defined in the `/etc/logadm.conf` file. This file includes log rotation entries for processes such as `syslogd`. For example, one entry in the `/etc/logadm.conf` file specifies that the `/var/log/syslog` file is rotated weekly unless the file is empty. The most recent `syslog` file becomes `syslog.0`, the next most recent becomes `syslog.1`, and so on. Eight previous `syslog` log files are kept.

The `/etc/logadm.conf` file also contains time stamps of when the last log rotation occurred.

You can use the `logadm` command to customize system logging and to add additional logging in the `/etc/logadm.conf` file as needed.

For example, to rotate the Apache access and error logs, use the following commands:

```
# logadm -w /var/apache/logs/access_log -s 100m
# logadm -w /var/apache/logs/error_log -s 10m
```

In this example, the Apache `access_log` file is rotated when it reaches 100 MB in size, with a `.0`, `.1`, (and so on) suffix, keeping 10 copies of the old `access_log` file. The `error_log` is rotated when it reaches 10 MB in size with the same suffixes and number of copies as the `access_log` file.

The `/etc/logadm.conf` entries for the preceding Apache log rotation examples look similar to the following:

```
# cat /etc/logadm.conf
.
.
.
/var/apache/logs/error_log -s 10m
/var/apache/logs/access_log -s 100m
```

For more information, see logadm(1M).

You can use the `logadm` command as superuser or by assuming an equivalent role (with Log Management rights). With role-based access control (RBAC), you can grant non-root users the privilege of maintaining log files by providing access to the `logadm` command.

For example, add the following entry to the /etc/user_attr file to grant user andy the ability to use the logadm command:

```
andy::::profiles=Log Management
```

Or, you can set up a role for log management by using the Solaris Management Console. For more information about setting up a role, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

Customizing System Message Logging

You can capture additional error messages that are generated by various system processes by modifying the /etc/syslog.conf file. By default, the /etc/syslog.conf file directs many system process messages to the /var/adm/messages files. Crash and boot messages are stored here as well. To view /var/adm messages, see "How to View System Messages" on page 224.

The /etc/syslog.conf file has two columns separated by tabs:

facility.level ... *action*

facility.level A *facility* or system source of the message or condition. May be a comma-separated listed of facilities. Facility values are listed in Table 15–1. A *level*, indicates the severity or priority of the condition being logged. Priority levels are listed in Table 15–2.

Do not put two entries for the same facility on the same line, if the entries are for different priorities. Putting a priority in the syslog file indicates that all messages of that all messages of that priority or higher are logged, with the last message taking precedence. For a given facility and level, syslogd matches all messages for that level and all higher levels.

action The action field indicates where the messages are forwarded.

The following example shows sample lines from a default /etc/syslog.conf file.

```
user.err                                          /dev/sysmsg
user.err                                          /var/adm/messages
user.alert                                        'root, operator'
user.emerg                                        *
```

This means the following user messages are automatically logged:

- User errors are printed to the console and also are logged to the /var/adm/messages file.
- User messages requiring immediate action (alert) are sent to the root and operator users.
- User emergency messages are sent to individual users.

Note – Placing entries on separate lines might cause messages to be logged out of order if a log target is specified more than once in the `/etc/syslog.conf` file. Note that you can specify multiple selectors in a single line entry, each separated by a semi-colon.

The most common error condition sources are shown in the following table. The most common priorities are shown in Table 15–2 in order of severity.

TABLE 15–1 Source Facilities for `syslog.conf` Messages

Source	Description
`kern`	The kernel
`auth`	Authentication
`daemon`	All daemons
`mail`	Mail system
`lp`	Spooling system
`user`	User processes

Note – The number of `syslog` facilities that can be activated in the `/etc/syslog.conf` file is unlimited.

TABLE 15–2 Priority Levels for `syslog.conf` Messages

Priority	Description
`emerg`	System emergencies
`alert`	Errors requiring immediate correction
`crit`	Critical errors
`err`	Other errors
`info`	Informational messages
`debug`	Output used for debugging
`none`	This setting doesn't log output

▼ How to Customize System Message Logging

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Edit the `/etc/syslog.conf` file, adding or changing message sources, priorities, and message locations according to the syntax described in `syslog.conf`(4).**

3. **Exit the file, saving the changes.**

Example 15–2 Customizing System Message Logging

This sample `/etc/syslog.conf` `user.emerg` facility sends user emergency messages to root *and* individual users.

```
user.emerg                                          'root, *'
```

Enabling Remote Console Messaging

The following new console features improve your ability to troubleshoot remote systems:

- The `consadm` command enables you to select a serial device as an *auxiliary* (or remote) console. Using the `consadm` command, a system administrator can configure one or more serial ports to display redirected console messages and to host `sulogin` sessions when the system transitions between run levels. This feature enables you to dial in to a serial port with a modem to monitor console messages and participate in `init` state transitions. (For more information, see `sulogin`(1M) and the step-by-step procedures that follow.)

  While you can log in to a system using a port configured as an auxiliary console, it is primarily an output device displaying information that is also displayed on the default console. If boot scripts or other applications read and write to and from the default console, the write output displays on all the auxiliary consoles, but the input is only read from the default console. (For more information on using the `consadm` command during an interactive login session, see "Using the `consadm` Command During an Interactive Login Session" on page 230.)

- Console output now consists of kernel and `syslog` messages written to a new pseudo device, `/dev/sysmsg`. In addition, `rc` script startup messages are written to `/dev/msglog`. Previously, all of these messages were written to `/dev/console`.

  Scripts that direct console output to `/dev/console` need to be changed to `/dev/msglog` if you want to see script messages displayed on the auxiliary consoles. Programs referencing `/dev/console` should be explicitly modified to use `syslog()` or `strlog()` if you want messages to be redirected to an auxiliary device.

- The consadm command runs a daemon to monitor auxiliary console devices. Any display device designated as an auxiliary console that disconnects, hangs up or loses carrier, is removed from the auxiliary console device list and is no longer active. Enabling one or more auxiliary consoles does not disable message display on the default console; messages continue to display on /dev/console.

Using Auxiliary Console Messaging During Run Level Transitions

Keep the following in mind when using auxiliary console messaging during run level transitions:

- Input cannot come from an auxiliary console if user input is expected for an rc script that is run when a system is booting. The input must come from the default console.
- The sulogin program, invoked by init to prompt for the superuser password when transitioning between run levels, has been modified to send the superuser password prompt to each auxiliary device in addition to the default console device.
- When the system is in single-user mode and one or more auxiliary consoles are enabled using the consadm command, a console login session runs on the first device to supply the correct superuser password to the sulogin prompt. When the correct password is received from a console device, sulogin disables input from all other console devices.
- A message is displayed on the default console and the other auxiliary consoles when one of the consoles assumes single-user privileges. This message indicates which device has become the console by accepting a correct superuser password. If there is a loss of carrier on the auxiliary console running the single-user shell, one of two actions might occur:
    - If the auxiliary console represents a system at run level 1, the system proceeds to the default run level.
    - If the auxiliary console represents a system at run level S, the system displays the ENTER RUN LEVEL (0-6, s or S): message on the device where the init s or shutdown command had been entered from the shell. If there isn't any carrier on that device either, you will have to reestablish carrier and enter the correct run level. The init or shutdown command will not redisplay the run-level prompt.
- If you are logged in to a system using a serial port, and an init or shutdown command is issued to transition to another run level, the login session is lost whether this device is the auxiliary console or not. This situation is identical to Solaris releases without auxiliary console capabilities.
- Once a device is selected as an auxiliary console using the consadm command, it remains the auxiliary console until the system is rebooted or the auxiliary console is unselected. However, the consadm command includes an option to set a device as the auxiliary console across system reboots. (See the following procedure for step-by-step instructions.)

Using the consadm Command During an Interactive Login Session

If you want to run an interactive login session by logging in to a system using a terminal that is connected to a serial port, and then using the consadm command to see the console messages from the terminal, note the following behavior:

- If you use the terminal for an interactive login session while the auxiliary console is active, the console messages are sent to the /dev/sysmsg or /dev/msglog devices.
- While you issue commands on the terminal, input goes to your interactive session and not to the default console (/dev/console).
- If you run the init command to change run levels, the remote console software kills your interactive session and runs the sulogin program. At this point, input is accepted only from the terminal and is treated like it's coming from a console device. This allows you to enter your password to the sulogin program as described in "Using Auxiliary Console Messaging During Run Level Transitions" on page 229.

  Then, if you enter the correct password on the (auxiliary) terminal, the auxiliary console runs an interactive sulogin session, locks out the default console and any competing auxiliary console. This means the terminal essentially functions as the system console.
- From here you can change to run level 3 or go to another run level. If you change run levels, sulogin runs again on all console devices. If you exit or specify that the system should come up to run level 3, then all auxiliary consoles lose their ability to provide input. They revert to being display devices for console messages.

  As the system is coming up, you must provide information to rc scripts on the default console device. After the system comes back up, the login program runs on the serial ports and you can log back into another interactive session. If you've designated the device to be an auxiliary console, you will continue to get console messages on your terminal, but all input from the terminal goes to your interactive session.

▼ How to Enable an Auxiliary (Remote) Console

The consadm daemon does not start monitoring the port until after you add the auxiliary console with the consadm command. As a security feature, console messages are only redirected until carrier drops, or the auxiliary console device is unselected. This means carrier must be established on the port before you can successfully use the consadm command.

For more information on enabling an auxiliary console, see the consadm(1m) man page.

1 **Log in to the system as superuser.**

2 **Enable the auxiliary console.**

```
# consadm -a devicename
```

3 **Verify that the current connection is the auxiliary console.**

```
# consadm
```

Example 15–3 Enabling an Auxiliary (Remote) Console

```
# consadm -a /dev/term/a
# consadm
 /dev/term/a
```

▼ How to Display a List of Auxiliary Consoles

1 **Log in to the system as superuser.**

2 **Select one of the following steps:**

a. **Display the list of auxiliary consoles.**

```
# consadm
/dev/term/a
```

b. **Display the list of persistent auxiliary consoles.**

```
# consadm -p
/dev/term/b
```

▼ How to Enable an Auxiliary (Remote) Console Across System Reboots

1 **Log in to the system as superuser.**

2 **Enable the auxiliary console across system reboots.**

```
# consadm -a -p devicename
```

This adds the device to the list of persistent auxiliary consoles.

3 **Verify that the device has been added to the list of persistent auxiliary consoles.**

```
# consadm
```

Example 15–4 Enabling an Auxiliary (Remote) Console Across System Reboots

```
# consadm -a -p /dev/term/a
# consadm
/dev/term/a
```

▼ How to Disable an Auxiliary (Remote) Console

1. **Log in to the system as superuser.**

2. **Select one of the following steps:**

   a. **Disable the auxiliary console.**

   ```
   # consadm -d devicename
   ```

   or

   b. **Disable the auxiliary console and remove it from the list of persistent auxiliary consoles.**

   ```
   # consadm -p -d devicename
   ```

3. **Verify that the auxiliary console has been disabled.**

   ```
   # consadm
   ```

Example 15–5 Disabling an Auxiliary (Remote) Console

```
# consadm -d /dev/term/a
# consadm
```

CHAPTER 16

Managing Core Files (Tasks)

This chapter describes how to manage core files with the coreadm command.

For information on the procedures associated with managing core files, see "Managing Core Files (Task Map)" on page 233.

Managing Core Files (Task Map)

Task	Description	For Instructions
1. Display the current core dump configuration	Display the current core dump configuration by using the coreadm command.	"How to Display the Current Core Dump Configuration" on page 236
2. Modify the core dump configuration	Modify the core dump configuration to do one of the following: Set a core file name pattern. Enable a per-process core file path. Enable a global core file path.	"How to Set a Core File Name Pattern" on page 237 "How to Enable a Per-Process Core File Path" on page 237 "How to Enable a Global Core File Path" on page 237
3. Examine a Core Dump File	Use the proc tools to view a core dump file.	"Examining Core Files" on page 238

Managing Core Files Overview

Core files are generated when a process or application terminates abnormally. Core files are managed with the `coreadm` command.

For example, you can use the `coreadm` command to configure a system so that all process core files are placed in a single system directory. This means it is easier to track problems by examining the core files in a specific directory whenever a Solaris process or daemon terminates abnormally.

Configurable Core File Paths

Two new configurable `core` file paths that can be enabled or disabled independently of each other are:

- A per-process core file path, which defaults to `core` and is enabled by default. If enabled, the per-process core file path causes a `core` file to be produced when the process terminates abnormally. The per-process path is inherited by a new process from its parent process.

  When generated, a per-process core file is owned by the owner of the process with read/write permissions for the owner. Only the owning user can view this file.
- A global core file path, which defaults to `core` and is disabled by default. If enabled, an *additional* core file with the same content as the per-process core file is produced by using the global core file path.

  When generated, a global core file is owned by superuser with read/write permissions for superuser only. Non-privileged users cannot view this file.

When a process terminates abnormally, it produces a core file in the current directory by default. If the global core file path is enabled, each abnormally terminating process might produce two files, one in the current working directory, and one in the global core file location.

By default, a `setuid` process does not produce core files using either the global or per-process path.

Expanded Core File Names

If a global core file directory is enabled, `core` files can be distinguished from one another by using the variables described in the following table.

Variable Name	Variable Definition
%d	Executable file directory name, up to a maximum of MAXPATHLEN characters
%f	Executable file name, up to a maximum of MAXCOMLEN characters
%g	Effective group ID
%m	Machine name (uname -m)
%n	System node name)uname -n)
%p	Process ID
%t	Decimal value of time(2)
%u	Effective user ID
%z	Name of the zone in which process is executed (zonename)
%%	Literal %

For example, if the global core file path is set to:

```
/var/core/core.%f.%p
```

and a sendmail process with PID 12345 terminates abnormally, it produces the following core file:

```
/var/core/core.sendmail.12345
```

Setting the Core File Name Pattern

You can set a core file name pattern on a global, zone, or per-process basis. In addition, you can set the per-process defaults that persist across a system reboot.

For example, the following coreadm command sets the default per-process core file pattern. This setting applies to all processes that have not explicitly overridden the default core file pattern. This setting persists across system reboots.

```
# coreadm -i /var/core/core.%f.%p
```

This coreadm command sets the per-process core file name pattern for any processes:

```
$ coreadm -p /var/core/core.%f.%p $$
```

The $$ symbols represent a placeholder for the process ID of the currently running shell. The per-process core file name pattern is inherited by all child processes.

Once a global or per-process core file name pattern is set, it must be enabled with the `coreadm -e` command. See the following procedures for more information.

You can set the core file name pattern for all processes run during a user's login session by putting the command in a user's `$HOME/.profile` or `.login` file.

Enabling `setuid` Programs to Produce Core Files

You can use the `coreadm` command to enable or disable `setuid` programs to produce core files for all system processes or on a per-process basis by setting the following paths:

- If the global `setuid` option is enabled, a global core file path allows all `setuid` programs on a system to produce `core` files.
- If the per-process `setuid` option is enable, a per-process core file path allows specific `setuid` processes to produce `core` files.

By default, both flags are disabled. For security reasons, the global core file path must be a full pathname, starting with a leading `/`. If superuser disables per-process core files, individual users cannot obtain core files.

The `setuid` core files are owned by superuser with read/write permissions for superuser only. Regular users cannot access them even if the process that produced the `setuid` core file was owned by an ordinary user.

For more information, see `coreadm`(1M).

How to Display the Current Core Dump Configuration

Use the `coreadm` command without any options to display the current core dump configuration.

```
$ coreadm
               global core file pattern:
     global core file content: default
       init core file pattern: core
       init core file content: default
            global core dumps: disabled
       per-process core dumps: enabled
      global setid core dumps: disabled
 per-process setid core dumps: disabled
     global core dump logging: disabled
```

▼ How to Set a Core File Name Pattern

● **Determine whether you want to set a per-process or global core file and select one of the following:**

a. **Set a per-process file name pattern.**

```
$ coreadm -p $HOME/corefiles/%f.%p $$
```

b. **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

c. **Set a global file name pattern.**

```
# coreadm -g /var/corefiles/%f.%p
```

▼ How to Enable a Per-Process Core File Path

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Enable a per-process core file path.**

```
# coreadm -e process
```

3 **Display the current process core file path to verify the configuration.**

```
$ coreadm $$
1180:   /home/kryten/corefiles/%f.%p
```

▼ How to Enable a Global Core File Path

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Enable a global core file path.**

```
# coreadm -e global -g /var/core/core.%f.%p
```

3 Display the current process core file path to verify the configuration.

```
# coreadm
       global core file pattern: /var/core/core.%f.%p
     global core file content: default
       init core file pattern: core
       init core file content: default
            global core dumps: enabled
       per-process core dumps: enabled
      global setid core dumps: disabled
 per-process setid core dumps: disabled
     global core dump logging: disabled
```

Troubleshooting Core File Problems

Error Message

```
NOTICE: 'set allow_setid_core = 1' in /etc/system is obsolete
NOTICE: Use the coreadm command instead of 'allow_setid_core'
```

Cause

You have an obsolete parameter that allows `setuid` core files in your `/etc/system` file.

Solution

Remove `allow_setid_core=1` from the `/etc/system` file. Then use the `coreadm` command to enable global `setuid` core file paths.

Examining Core Files

Some of the proc tools have been enhanced to examine process core files as well as live processes. The proc tools are utilities that can manipulate features of the `/proc` file system.

The `/usr/proc/bin/pstack`, `pmap`, `pldd`, `pflags`, and `pcred` tools can now be applied to core files by specifying the name of the core file on the command line, similar to the way you specify a process ID to these commands.

For more information on using proc tools to examine core files, see `proc(1)`.

EXAMPLE 16–1 Examining Core Files With proc Tools

```
$ ./a.out
Segmentation Fault(coredump)
$ /usr/proc/bin/pstack ./core
core './core' of 19305: ./a.out
 000108c4 main     (1, ffbef5cc, ffbef5d4, 20800, 0, 0) + 1c
```

EXAMPLE 16–1 Examining Core Files With proc Tools *(Continued)*

```
 00010880 _start   (0, 0, 0, 0, 0, 0) + b8
```

◆ ◆ ◆ CHAPTER 17

Managing System Crash Information (Tasks)

This chapter describes how to manage system crash information in the Solaris Operating System.

For information on the procedures associated with managing system crash information, see "Managing System Crash Information (Task Map)" on page 241.

Managing System Crash Information (Task Map)

The following task map identifies the procedures needed to manage system crash information.

Task	Description	For Instructions
1. Display the current crash dump configuration.	Display the current crash dump configuration by using the `dumpadm` command.	"How to Display the Current Crash Dump Configuration" on page 245
2. Modify the crash dump configuration.	Use the `dumpadm` command to specify the type of data to dump, whether or not the system will use a dedicated dump device, the directory for saving crash dump files, and the amount of space that must remain available after crash dump files are written.	"How to Modify a Crash Dump Configuration" on page 246
3. Examine a crash dump file.	Use the `mdb` command to view crash dump files.	"How to Examine a Crash Dump" on page 247
4. (Optional) Recover from a full crash dump directory.	The system crashes, but no room is available in the `savecore` directory, and you want to save some critical system crash dump information.	"How to Recover From a Full Crash Dump Directory (Optional)" on page 248

Task	Description	For Instructions
5. (Optional) Disable or enable the saving of crash dump files.	Use the `dumpadm` command to disable or enable the saving the crash dump files. Saving crash dump files is enabled by default.	"How to Disable or Enable Saving Crash Dumps" on page 248

System Crashes (Overview)

System crashes can occur due to hardware malfunctions, I/O problems, and software errors. If the system crashes, it will display an error message on the console, and then write a copy of its physical memory to the dump device. The system will then reboot automatically. When the system reboots, the `savecore` command is executed to retrieve the data from the dump device and write the saved crash dump to your `savecore` directory. The saved crash dump files provide invaluable information to your support provider to aid in diagnosing the problem.

ZFS Support for Swap Area and Dump Devices

If you select a ZFS™ root file system during an initial software installation, or if you use Solaris Live Upgrade to migrate from a UFS root file system to a ZFS root file system, a swap area is created on a ZFS volume in the ZFS root pool. Swap volume size is calculated as half the size of physical memory, but no more than 2 Gbytes and no less than 512 Mbytes. Dump volume size is calculated by the kernel based on `dumpadm` information and the size of physical memory. You can adjust the sizes of your swap and dump volumes in a JumpStart profile or during an initial installation to sizes of your choosing as long as the new sizes support system operation. For more information, see "ZFS Support for Swap and Dump Devices" in *Solaris ZFS Administration Guide*.

If you need to modify your ZFS swap area or dump area after installation, use the `swap` or `dumpadm` commands, as in previous Solaris releases.

For information about managing dump devices in this document, see "Managing System Crash Dump Information" on page 245.

x86: System Crashes in the GRUB Boot Environment

If a system crash occurs on an x86 based system in the GRUB boot environment, it is possible that the SMF service that manages the GRUB boot archive, `svc:/system/boot-archive:default`, might fail on the next system reboot. To troubleshoot this type of problem, see "x86: What to Do if the SMF Boot Archive Service Fails During a System Reboot" on page 255. For more information on GRUB based booting, see "Booting an x86 Based System by Using GRUB (Task Map)" in *System Administration Guide: Basic Administration*.

System Crash Dump Files

The `savecore` command runs automatically after a system crash to retrieve the crash dump information from the dump device and writes a pair of files called `unix`.*X* and `vmcore`.*X*, where *X* identifies the dump sequence number. Together, these files represent the saved system crash dump information.

Crash dump files are sometimes confused with *core* files, which are images of user applications that are written when the application terminates abnormally.

Crash dump files are saved in a predetermined directory, which by default, is `/var/crash/`*hostname*. In previous Solaris releases, crash dump files were overwritten when a system rebooted, unless you manually enabled the system to save the images of physical memory in a crash dump file. Now, the saving of crash dump files is enabled by default.

System crash information is managed with the `dumpadm` command. For more information, see "The `dumpadm` Command" on page 243.

Saving Crash Dumps

You can examine the control structures, active tables, memory images of a live or crashed system kernel, and other information about the operation of the kernel by using the `mdb` utility. Using `mdb` to its full potential requires a detailed knowledge of the kernel, and is beyond the scope of this manual. For information on using this utility, see the `mdb(1)` man page.

Additionally, crash dumps saved by `savecore` can be useful to send to a customer service representative for analysis of why the system is crashing.

The `dumpadm` Command

Use the `dumpadm` command to manage system crash dump information in the Solaris Operating System.

- The `dumpadm` command enables you to configure crash dumps of the operating system. The `dumpadm` configuration parameters include the dump content, dump device, and the directory in which crash dump files are saved.
- Dump data is stored in compressed format on the dump device. Kernel crash dump images can be as big as 4 Gbytes or more. Compressing the data means faster dumping and less disk space needed for the dump device.
- Saving crash dump files is run in the background when a dedicated dump device, not the swap area, is part of the dump configuration. This means a booting system does not wait for the `savecore` command to complete before going to the next step. On large memory systems, the system can be available before `savecore` completes.

- System crash dump files, generated by the savecore command, are saved by default.
- The savecore -L command is a new feature which enables you to get a crash dump of the live running the Solaris OS. This command is intended for troubleshooting a running system by taking a snapshot of memory during some bad state, such as a transient performance problem or service outage. If the system is up and you can still run some commands, you can execute the savecore -L command to save a snapshot of the system to the dump device, and then immediately write out the crash dump files to your savecore directory. Because the system is still running, you can only use the savecore -L command if you have configured a dedicated dump device.

The following table describes dumpadm's configuration parameters.

Dump Parameter	Description
dump device	The device that stores dump data temporarily as the system crashes. When the dump device is not the swap area, savecore runs in the background, which speeds up the boot process.
savecore directory	The directory that stores system crash dump files.
dump content	Type of memory data to dump.
minimum free space	Minimum amount of free space required in the savecore directory after saving crash dump files. If no minimum free space has been configured, the default is one Mbyte.

For more information, see dumpadm(1M).

Dump configuration parameters are managed by the dumpadm command.

How the dumpadm Command Works

During system startup, the dumpadm command is invoked by the svc:/system/dumpadm:default service to configure crash dumps parameters.

Specifically, dumpadm initializes the dump device and the dump content through the /dev/dump interface.

After the dump configuration is complete, the savecore script looks for the location of the crash dump file directory. Then, savecore is invoked to check for crash dumps and check the content of the minfree file in the crash dump directory.

Dump Devices and Volume Managers

Do not configure a dedicated dump device that is under the control of volume management product such as Solaris Volume Manager for accessibility and performance reasons. You can keep your swap areas under the control of Solaris Volume Manager and this is a recommend practice, but keep your dump device separate.

Managing System Crash Dump Information

Keep the following key points in mind when you are working with system crash information:

- You must be superuser or assume an equivalent role to access and manage system crash information.
- Do not disable the option of saving system crash dumps. System crash dump files provide an invaluable way to determine what is causing the system to crash.
- Do not remove important system crash information until it has been sent to your customer service representative.

▼ How to Display the Current Crash Dump Configuration

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2. **Display the current crash dump configuration.**

   ```
   # dumpadm
   Dump content: kernel pages
   Dump device: /dev/dsk/c0t3d0s1 (swap)
   Savecore directory: /var/crash/venus
   Savecore enabled: yes
   ```

   The preceding example output means:

   - The dump content is kernel memory pages.
   - Kernel memory will be dumped on a swap device, `/dev/dsk/c0t3d0s1`. You can identify all your swap areas with the `swap -l` command.
   - System crash dump files will be written in the `/var/crash/venus` directory.
   - Saving crash dump files is enabled.

▼ How to Modify a Crash Dump Configuration

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Identify the current crash dump configuration.**

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/pluto
  Savecore enabled: yes
```

This output identifies the default dump configuration for a system running the Solaris 10 release.

3 **Modify the crash dump configuration.**

`# dumpadm -c` *content* `-d` *dump-device* `-m` *nnnk* | *nnnm* | *nnn%* `-n -s` *savecore-dir*

`-c` *content*
Specifies the type of data to dump. Use `kernel` to dump of all kernel memory, `all` to dump all of memory, or `curproc`, to dump kernel memory and the memory pages of the process whose thread was executing when the crash occurred. The default dump content is kernel memory.

`-d` *dump-device*
Specifies the device that stores dump data temporarily as the system crashes. The primary swap device is the default dump device.

`-m` *nnnk* | *nnnm* | *nnn%*
Specifies the minimum free disk space for saving crash dump files by creating a `minfree` file in the current `savecore` directory. This parameter can be specified in Kbytes (`nnnk`), Mbytes (`nnnm`) or file system size percentage (`nnn%`). The `savecore` command consults this file prior to writing the crash dump files. If writing the crash dump files, based on their size, would decrease the amount of free space below the `minfree` threshold, the dump files are not written and an error message is logged. For information on recovering from this scenario, see "How to Recover From a Full Crash Dump Directory (Optional)" on page 248.

`-n`
Specifies that `savecore` should not be run when the system reboots. This dump configuration is not recommended. If system crash information is written to the swap device, and `savecore` is not enabled, the crash dump information is overwritten when the system begins to swap.

`-s` Specifies an alternate directory for storing crash dump files. The default directory is `/var/crash/hostname` where `hostname` is the output of the `uname -n` command.

Example 17–1 Modifying a Crash Dump Configuration

In this example, all of memory is dumped to the dedicated dump device, `/dev/dsk/c0t1d0s1`, and the minimum free space that must be available after the crash dump files are saved is 10% of the file system space.

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/pluto
  Savecore enabled: yes
 # dumpadm -c all -d /dev/dsk/c0t1d0s1 -m 10%
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: yes
```

▼ How to Examine a Crash Dump

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Examine a crash dump by using the `mdb` utility.**

```
# /usr/bin/mdb [-k] crashdump-file
```

`-k` Specifies kernel debugging mode by assuming the file is an operating system crash dump file.

crashdump-file Specifies the operating system crash dump file.

3 **Display crash status information.**

```
# /usr/bin/mdb file-name
> ::status
   .
   .
   .
> ::system
   .
```

```
.
.
```

Example 17–2 Examining a Crash Dump

The following example shows sample output from the `mdb` utility, which includes system information and identifies the tunables that are set in this system's `/etc/system` file.

```
# /usr/bin/mdb -k unix.0
Loading modules: [ unix krtld genunix ip nfs ipc ptm ]
> ::status
debugging crash dump /dev/mem (64-bit) from ozlo
operating system: 5.10 Generic (sun4u)
> ::system
set ufs_ninode=0x9c40 [0t40000]
set ncsize=0x4e20 [0t20000]
set pt_cnt=0x400 [0t1024]
```

▼ How to Recover From a Full Crash Dump Directory (Optional)

In this scenario, the system crashes but no room is left in the `savecore` directory, and you want to save some critical system crash dump information.

1. **Log in as superuser or assume an equivalent role after the system reboots.**

2. **Clear out the savecore directory, usually `/var/crash/`*hostname*, by removing existing crash dump files that have already been sent to your service provider. Or, run the `savecore` command and specify an alternate directory that has sufficient disk space. See the next step.**

3. **Manually run the `savecore` command and if necessary, specify an alternate savecore directory.**

   ```
   # savecore [ directory ]
   ```

▼ How to Disable or Enable Saving Crash Dumps

1. **Become superuser or assume an equivalent role.**

   Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Disable or enable the saving of crash dumps on your system.**

```
# dumpadm -n | -y
```

Example 17–3 Disabling the Saving of Crash Dumps

This example illustrates how to disable the saving of crash dumps on your system.

```
# dumpadm -n
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: no
```

Example 17–4 Enabling the Saving of Crash Dumps

This example illustrates how to enable the saving of crash dump on your system.

```
# dumpadm -y
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: yes
```

CHAPTER 18

Troubleshooting Miscellaneous Software Problems (Tasks)

This chapter describes miscellaneous software problems that might occur occasionally and are relatively easy to fix. Troubleshooting miscellaneous software problems includes solving problems that aren't related to a specific software application or topic, such as unsuccessful reboots and full file systems. Resolving these problems are described in the following sections.

This is a list of the information in this chapter.

What to Do if Rebooting Fails

If the system does not reboot completely, or if it reboots and then crashes again, there might be a software or hardware problem that is preventing the system from booting successfully.

Cause of System Not Booting	How to Fix the Problem
The system can't find `/platform/‘uname -m‘/kernel/unix`.	You may need to change the `boot-device` setting in the PROM on a SPARC based system. For information on changing the default boot device, see "How to Change the Default Boot Device by Using the Boot PROM" in *System Administration Guide: Basic Administration*.

Cause of System Not Booting	How to Fix the Problem
Solaris 10: There is no default boot device on an x86 based system. The message displayed is: `Not a UFS filesystem.`	**Solaris 10:** Boot the system by using the Configuration Assistant/boot diskette and select the disk from which to boot.
Solaris 10 1/06: The GRUB boot archive has become corrupted. Or, the SMF boot archive service has failed. An error message is displayed if you run the `svcs -x` command.	**Solaris 10 1/06:** Boot the failsafe archive.
There's an invalid entry in the `/etc/passwd` file.	For information on recovering from an invalid `passwd` file, see Chapter 12, "Booting a Solaris System (Tasks)," in *System Administration Guide: Basic Administration*.
There's a hardware problem with a disk or another device.	Check the hardware connections: ■ Make sure the equipment is plugged in. ■ Make sure all the switches are set properly. ■ Look at all the connectors and cables, including the Ethernet cables. ■ If all this fails, turn off the power to the system, wait 10 to 20 seconds, and then turn on the power again.

If none of the above suggestions solve the problem, contact your local service provider.

What to Do if You Forgot Root Password

If you forget the root password and you cannot log into the system, you will have to do the following:

- Stop the system by using the keyboard stop sequence.
- **Starting with Solaris 10 1/06 release:** On x86 based systems, boot the system in the Solaris failsafe archive.
- **Solaris 10:** Boot the system from a boot server or an install server, or from a local CD-ROM.
- Mount the root (`/`) file system.
- Remove the root password from the `/etc/shadow` file.
- Reboot the system.
- Log in and set root's password.

These procedures are fully described in Chapter 12, "Booting a Solaris System (Tasks)," in *System Administration Guide: Basic Administration*

Note – GRUB based booting is not available on SPARC based systems in this Solaris release.

The following examples describe how to recover from a forgotten root password on both SPARC and x86 based systems.

EXAMPLE 18–1 SPARC: What to Do if You Forgot Root Password

The following example shows how to recover when you forget the root password by booting from the network. This example assumes that the boot server is already available. Be sure to apply a new root password after the system has rebooted.

```
(Use keyboard abort sequence--Press Stop A keys to stop the system)
ok boot net -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi shadow
(Remove root’s encrypted password string)
# cd /
# umount /a
# init 6
```

EXAMPLE 18–2 x86: Performing a GRUB Based Boot When You Have Forgotten the Root Password

This example assumes that the boot server is already available. Be sure to apply a new root password after the system has rebooted.

```
GNU GRUB  version 0.95  (637K lower / 3144640K upper memory)
 +-------------------------------------------------------------------+
| be1
| be1 failsafe
| be3
| be3 failsafe
| be2
| be2 failfafe
  +------------------------------------------------------------------+
      Use the ^ and v keys to select which entry is highlighted.
      Press enter to boot the selected OS, ’e’ to edit the
      commands before booting, or ’c’ for a command-line.

Searching for installed OS instances...

    An out of sync boot archive was detected on /dev/dsk/c0t0d0s0.
    The boot archive is a cache of files used during boot and
```

EXAMPLE 18–2 x86: Performing a GRUB Based Boot When You Have Forgotten the Root Password *(Continued)*

```
    should be kept in sync to ensure proper system operation.

    Do you wish to automatically update this boot archive? [y,n,?] n
Searching for installed OS instances...

Multiple OS instances were found. To check and mount one of them
read-write under /a, select it from the following list. To not mount
any, select ’q’.

  1  pool10:13292304648356142148     ROOT/be10
  2  rpool:14465159259155950256      ROOT/be01

Please select a device to be mounted (q for none) [?,??,q]: 1
mounting /dev/dsk/c0t0d0s0 on /a
starting shell.
      .
      .
      .
# cd /a/etc
# vi shadow
(Remove root’s encrypted password string)
# cd /
# umount /a
# reboot
```

EXAMPLE 18–3 x86: Booting a System When You Have Forgotten the Root Password

Solaris 10: The following example shows how to recover when you forget root's password by booting from the network. This example assumes that the boot server is already available. Be sure to apply a new root password after the system has rebooted.

```
Press any key to reboot.
Resetting...
.
.
.
Initializing system
Please wait...


                     <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
```

EXAMPLE 18–3 x86: Booting a System When You Have Forgotten the Root Password *(Continued)*

```
Type    b [file-name] [boot-flags] <ENTER>     to boot with options
or      i <ENTER>                              to enter boot interpreter
or      <ENTER>                                to boot with defaults

                  <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b -s
SunOS Release 5.10 Version amd64-gate-2004-09-30 32-bit
Copyright 1983-2004 Sun Microsystems, Inc.  All rights reserved.
Use is subject to license terms.
DEBUG enabled
Booting to milestone "milestone/single-user:default".
Hostname: venus
NIS domain name is example.com
Requesting System Maintenance Mode
SINGLE USER MODE

Root password for system maintenance (control-d to bypass): xxxxxx
Entering System Maintenance Mode
.
.
.
# mount /dev/dsk/c0t0d0s0 /a
      .
      .
      .
# cd /a/etc
# vi shadow
(Remove root's encrypted password string)
# cd /
# umount /a
# init 6
```

x86: What to Do if the SMF Boot Archive Service Fails During a System Reboot

Solaris 10 1/06: If the system crashes, the boot archive SMF service, `svc:/system/boot-archive:default`, might fail when the system is rebooted. If the boot archive service has failed, a message similar to the following is displayed when you run the `svcs -x` command:

```
svc:/system/boot-archive:default (check boot archive content)
 State: maintenance since Fri Jun 03 10:24:52 2005
Reason: Start method exited with $SMF_EXIT_ERR_FATAL.
```

```
   See: http://sun.com/msg/SMF-8000-KS
   See: /etc/svc/volatile/system-boot-archive:default.log
Impact: 48 dependent services are not running.  (Use -v for list.)

svc:/network/rpc/gss:default (Generic Security Service)
 State: uninitialized since Fri Jun 03 10:24:51 2005
Reason: Restarter svc:/network/inetd:default is not running.
   See: http://sun.com/msg/SMF-8000-5H
   See: gssd(1M)
Impact: 10 dependent services are not running.  (Use -v for list.)

svc:/application/print/server:default (LP print server)
 State: disabled since Fri Jun 03 10:24:51 2005
Reason: Disabled by an administrator.
   See: http://sun.com/msg/SMF-8000-05
   See: lpsched(1M)
Impact: 1 dependent service is not running.  (Use -v for list.)
```

To correct the problem, take the following action:

1. Reboot the system and select the Solaris failsafe archive option from the GRUB boot menu.
2. Answer y when prompted by the system to rebuild the boot archive.

   After the boot archive is rebuilt, the system is ready to boot.
3. To continue booting, clear the SMF boot archive service by using the following command.

```
# svcadm clear boot-archive
```

Note that you must become superuser or the equivalent to run this command.

For more information on rebuilding the GRUB boot archive, see “How to Boot the Failsafe Archive on an x86 Based System by Using GRUB” in *System Administration Guide: Basic Administration* and the `bootadm(1M)` man page.

What to Do if a System Hangs

A system can freeze or hang rather than crash completely if some software process is stuck. Follow these steps to recover from a hung system.

1. Determine whether the system is running a window environment and follow these suggestions. If these suggestions don't solve the problem, go to step 2.
   - Make sure the pointer is in the window where you are typing the commands.
   - Press Control-q in case the user accidentally pressed Control-s, which freezes the screen. Control-s freezes only the window, not the entire screen. If a window is frozen, try using another window.

- If possible, log in remotely from another system on the network. Use the `pgrep` command to look for the hung process. If it looks like the window system is hung, identify the process and kill it.

2. Press Control-\ to force a "quit" in the running program and (probably) write out a `core` file.
3. Press Control-c to interrupt the program that might be running.
4. Log in remotely and attempt to identify and kill the process that is hanging the system.
5. Log in remotely, become superuser or assume an equivalent role and reboot the system.
6. If the system still does not respond, force a crash dump and reboot. For information on forcing a crash dump and booting, see "Forcing a Crash Dump and Reboot of the System" in *System Administration Guide: Basic Administration*.
7. If the system still does not respond, turn the power off, wait a minute or so, then turn the power back on.
8. If you cannot get the system to respond at all, contact your local service provider for help.

What to Do if a File System Fills Up

When the root (`/`) file system or any other file system fills up, you will see the following message in the console window:

```
.... file system full
```

There are several reasons why a file system fills up. The following sections describe several scenarios for recovering from a full file system. For information on routinely cleaning out old and unused files to prevent full file systems, see Chapter 6, "Managing Disk Use (Tasks)."

File System Fills Up Because a Large File or Directory Was Created

Reason Error Occurred	How to Fix the Problem
Someone accidentally copied a file or directory to the wrong location. This also happens when an application crashes and writes a large `core` file into the file system.	Log in as superuser or assume an equivalent role and use the `ls -tl` command in the specific file system to identify which large file is newly created and remove it. For information on removing `core` files, see "How to Find and Delete `core` Files" on page 91.

A TMPFS File System is Full Because the System Ran Out of Memory

Reason Error Occurred	How to Fix the Problem
This can occur if TMPFS is trying to write more than it is allowed or some current processes are using a lot of memory.	For information on recovering from tmpfs-related error messages, see the tmpfs(7FS) man page.

What to Do if File ACLs Are Lost After Copy or Restore

Reason Error Occurred	How to Fix the Problem
If files or directories with ACLs are copied or restored into the /tmp directory, the ACL attributes are lost. The /tmp directory is usually mounted as a temporary file system, which doesn't support UFS file system attributes such as ACLs.	Copy or restore files into the /var/tmp directory instead.

Troubleshooting Backup Problems

This section describes some basic troubleshooting techniques to use when backing up and restoring data.

The root (/) File System Fills Up After You Back Up a File System

You back up a file system, and the root (/) file system fills up. Nothing is written to the media, and the ufsdump command prompts you to insert the second volume of media.

Reason Error Occurred	How to Fix the Problem
If you used an invalid destination device name with the `-f` option, the `ufsdump` command wrote to a file in the `/dev` directory of the root (`/`) file system, filling it up. For example, if you typed `/dev/rmt/st0` instead of `/dev/rmt/0`, the backup file `/dev/rmt/st0` was created on the disk rather than being sent to the tape drive.	Use the `ls -tl` command in the `/dev` directory to identify which file is newly created and abnormally large, and remove it.

Make Sure the Backup and Restore Commands Match

You can only use the `ufsrestore` command to restore files backed up with the `ufsdump` command. If you back up with the `tar` command, restore with the `tar` command. If you use the `ufsrestore` command to restore a tape that was written with another command, an error message tells you that the tape is not in `ufsdump` format.

Check to Make Sure You Have the Right Current Directory

It is easy to restore files to the wrong location. Because the `ufsdump` command always copies files with full path names relative to the root of the file system, you should usually change to the root directory of the file system before running the `ufsrestore` command. If you change to a lower-level directory, after you restore the files you will see a complete file tree created under that directory.

Interactive Commands

When you use the interactive command, a `ufsrestore>` prompt is displayed, as shown in this example:

```
# ufsrestore ivf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump   date: Fri Jan 30 10:13:46 2004
Dumped from: the epoch
Level 0 dump of /export/home on starbug:/dev/dsk/c0t0d0s7
Label: none
Extract directories from tape
Initialize symbol table.
ufsrestore >
```

At the `ufsrestore>` prompt, you can use the commands listed on Chapter 27, "UFS Backup and Restore Commands (Reference)," in *System Administration Guide: Devices and File Systems* to find files, create a list of files to be restored, and restore them.

Troubleshooting Common Agent Container Problems in the Solaris OS

This section addresses problems that you might encounter with the common agent container shared component. In this Solaris release, the common agent container Java program is included in the Solaris OS. The program implements a container for Java management applications. Typically, the container is not visible to the user.

The following are potential problems:

- Port number conflicts
- Compromised security for the superuser password

Port Number Conflicts

The common agent container occupies the following port numbers by default:

- JMX port (TCP) = `11162`
- SNMPAdaptor port (UDP) = `11161`
- SNMPAdaptor port for traps (UDP) = `11162`
- Commandstream Adaptor port (TCP) = `11163`
- RMI connector port (TCP) = `11164`

Note – If you are troubleshooting an installation of Sun Cluster, the port assignments are different.

If your installation already reserves any of these port numbers, change the port numbers that are occupied by the common agent container, as described in the following procedure.

▼ How to Check Port Numbers

This procedure shows you how to verify the Solaris port.

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Stop the common agent container management daemon.**

```
# /usr/sbin/cacaoadm stop
```

3 **Change the port numbs by using the following syntax:**

```
# /usr/sbin/cacaoadm set-param param=value
```

For example, to change the port occupied by the `SNMPAdaptor` from the default of 11161 to 11165, type:

```
# /usr/sbin/cacaoadm set-param snmp-adaptor-port=11165
```

4 **Restart the common agent container management daemon.**

```
# /usr/sbin/cacaoadm start
```

Compromised Security for Superuser Password

It might be necessary to regenerate security keys on a host that is running the Java ES. For example, if there is a risk that a superuser password has been exposed or compromised, you should regenerate the security keys. The keys that are used by the common agent container services are stored in `/etc/cacao/instances/`*instance-name*`/security` directory. The following task shows you how to generate security keys for the Solaris OS.

▼ How to Generate Security Keys for the Solaris OS

1 **Become superuser or assume an equivalent role.**

Roles contain authorizations and privileged commands. For more information about roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

2 **Stop the common agent container management daemon.**

```
# /usr/sbin/cacaoadm stop
```

3 **Regenerate the security keys.**

```
# /usr/sbin/cacaoadm create-keys --force
```

4 **Restart the common agent container management daemon.**

```
# /usr/sbin/cacaoadm start
```

Note – For the Sun Cluster software, you must propagate this change across all nodes in the cluster.

◆ ◆ ◆ CHAPTER 19

Troubleshooting File Access Problems (Tasks)

This chapter provides information on resolving file access problems such as those related to incorrect permissions and search paths.

This is a list of troubleshooting topics in this chapter.

Users frequently experience problems, and call on a system administrator for help, because they cannot access a program, a file, or a directory that they could previously use.

Whenever you encounter such a problem, investigate one of three areas:

- The user's search path may have been changed, or the directories in the search path may not be in the proper order.
- The file or directory may not have the proper permissions or ownership.
- The configuration of a system accessed over the network may have changed.

This chapter briefly describes how to recognize problems in each of these three areas and suggests possible solutions.

Solving Problems With Search Paths (`Command not found`)

A message of `Command not found` indicates one of the following:

- The command is not available on the system.
- The command directory is not in the search path.

To fix a search path problem, you need to know the pathname of the directory where the command is stored.

If the wrong version of the command is found, a directory that has a command of the same name is in the search path. In this case, the proper directory may be later in the search path or may not be present at all.

You can display your current search path by using the `echo $PATH` command. For example:

```
$ echo $PATH
/home/kryten/bin:/sbin:/usr/sbin:/usr/bin:/usr/dt:/usr/dist/exe
```

Use the `which` command to determine whether you are running the wrong version of the command. For example:

```
$ which acroread
/usr/doctools/bin/acroread
```

Note – The `which` command looks in the `.cshrc` file for path information. The `which` command might give misleading results if you execute it from the Bourne or Korn shell and you have a `.cshrc` file than contains aliases for the `which` command. To ensure accurate results, use the `which` command in a C shell, or, in the Korn shell, use the `whence` command.

How to Diagnose and Correct Search Path Problems

1 **Display the current search path to verify that the directory for the command is not in your path or that it isn't misspelled.**

```
$ echo $PATH
```

2 **Check the following:**

- Is the search path correct?
- Is the search path listed before other search paths where another version of the command is found?
- Is the command in one of the search paths?

If the path needs correction, go to step 3. Otherwise, go to step 4.

3 **Add the path to the appropriate file, as shown in this table.**

Shell	File	Syntax	Notes
Bourne and Korn	`$HOME/.profile`	`$ PATH=`*$HOME/bin:/sbin:/usr/local /bin …* `$ export PATH`	A colon separates path names.

Shell	File	Syntax	Notes

C	`$HOME/.cshrc`
	or
	`$HOME/.login`
		hostname% `set path=(`*~bin /sbin /usr/local/bin …)*	A blank space separates path names.

4 **Activate the new path as follows:**

Shell	File Where Path Is Located	Use this Command to Activate The Path
Bourne and Korn	`.profile`	`$` **`. ./.profile`**
C	`.cshrc`	*hostname%* **`source .cshrc`**
	`.login`	*hostname%* **`source .login`**

5 **Verify the new path.**

```
$ which command
```

Example 19–1 Diagnosing and Correcting Search Path Problems

This example shows that the `mytool` executable is not in any of the directories in the search path using the `which` command.

```
venus% mytool
mytool: Command not found
venus% which mytool
no mytool in /sbin /usr/sbin /usr/bin /etc /home/ignatz/bin  .
venus% echo $PATH
/sbin /usr/sbin /usr/bin /etc /home/ignatz/bin
venus% vi ~/.cshrc
(Add appropriate command directory to the search path)
venus% source .cshrc
venus% mytool
```

If you cannot find a command, look at the man page for its directory path. For example, if you cannot find the `lpsched` command (the `lp` printer daemon), the `lpsched(1M)` man page tells you the path is `/usr/lib/lp/lpsched`.

Solving File Access Problems

When users cannot access files or directories that they previously could access, the permissions or ownership of the files or directories probably has changed.

Changing File and Group Ownerships

Frequently, file and directory ownerships change because someone edited the files as superuser. When you create home directories for new users, be sure to make the user the owner of the dot (`.`) file in the home directory. When users do not own "`.`" they cannot create files in their own home directory.

Access problems can also arise when the group ownership changes or when a group of which a user is a member is deleted from the `/etc/group` database.

For information about how to change the permissions or ownership of a file that you are having problems accessing, see Chapter 6, "Controlling Access to Files (Tasks)," in *System Administration Guide: Security Services*.

Recognizing Problems With Network Access

If users have problems using the `rcp` remote copy command to copy files over the network, the directories and files on the remote system may have restricted access by setting permissions. Another possible source of trouble is that the remote system and the local system are not configured to allow access.

See "Strategies for NFS Troubleshooting" in *System Administration Guide: Network Services* for information about problems with network access and problems with accessing systems through AutoFS.

CHAPTER 20

Resolving UFS File System Inconsistencies (Tasks)

This chapter describes the fsck error messages and the possible responses you can make to resolve the error messages.

This is a list of the information in this chapter:

- "General fsck Error Messages" on page 269
- "Initialization Phase fsck Messages" on page 271
- "Phase 1: Check Blocks and Sizes Messages" on page 274
- "Phase 1B: Rescan for More DUPS Messages" on page 279
- **Solaris 10:** "Solaris 10: Phase 1B: Rescan for More DUPS Messages" on page 278
- "Phase 2: Check Path Names Messages" on page 279
- "Phase 3: Check Connectivity Messages" on page 286
- "Phase 4: Check Reference Counts Messages" on page 288
- "Phase 5: Check Cylinder Groups Messages" on page 291
- **Solaris 10:** "Phase 5: Check Cylinder Groups Messages" on page 292
- "fsck Summary Messages" on page 293
- **Solaris 10:** "Cleanup Phase Messages" on page 294

For information about the fsck command and how to use it to check file system integrity, see Chapter 21, "Checking UFS File System Consistency (Tasks)," in *System Administration Guide: Devices and File Systems*.

New fsck Error Messages

Solaris 10 6/06: In this Solaris release, error messages that are displayed when you run the fsck command have changed. This section includes the revised fsck error messages. If you are not running at least the Solaris 10 6/06 release, refer to the error messages in this chapter that are labeled "Solaris 10". For a detailed description of all the fsck improvements in the current Solaris release, see *System Administration Guide: Devices and File Systems*.

fsck Error Messages

Normally, the fsck command is run non-interactively to *preen* the file systems after an abrupt system halt in which the latest file system changes were not written to disk. Preening automatically fixes any basic file system inconsistencies and does not try to repair more serious errors. While preening a file system, the fsck command fixes the inconsistencies it expects from such an abrupt halt. For more serious conditions, the command reports the error and terminates.

When you run the fsck command interactively, it reports each inconsistency found and fixes innocuous errors. However, for more serious errors, the command reports the inconsistency and prompts you to choose a response. When you run the fsck command with the -y or -n options, your response is predefined as yes or no to the default response suggested by the fsck command for each error condition.

Some corrective actions will result in some loss of data. The amount and severity of data loss might be determined from the fsck diagnostic output.

The fsck command is a multipass file system check program. Each pass invokes a different phase of the fsck command with different sets of messages. After initialization, the fsck command performs successive passes over each file system, checking blocks and sizes, path names, connectivity, reference counts, and the map of free blocks (possibly rebuilding it). It also performs some cleanup.

The phases (passes) performed by the UFS version of the fsck command are:

- Initialization
- Phase 1 – Check Blocks and Sizes
- Phase 2a – Check Duplicated Names
- Phase 2b – Check Pathnames
- Phase 3 – Check Connectivity
- Phase 3b – Verify Shadows/ACLs
- Phase 4 – Check Reference Counts
- Phase 5 – Check Cylinder Groups

The next sections describe the error conditions that might be detected in each phase, the messages and prompts that result, and possible responses you can make.

Messages that might appear in more than one phase are described in “General fsck Error Messages” on page 269. Otherwise, messages are organized alphabetically by the phases in which they occur.

The following table lists many of the abbreviations included in the fsck error messages.

TABLE 20–1 Error Message Abbreviations

Abbreviation	Meaning
BLK	Block number
DUP	Duplicate block number
DIR	Directory name
CG	Cylinder group
MTIME	Time file was last modified
UNREF	Unreferenced

Many of the messages also include variable fields, such as inode numbers, which are represented in this book by an italicized term, such as *inode-number*. For example, this screen message:

```
INCORRECT BLOCK COUNT I=2529
```

is shown as follows:

INCORRECT BLOCK COUNT I=*inode-number*

General fsck Error Messages

The error messages in this section might be displayed in any phase after initialization. Although they offer the option to continue, it is generally best to regard them as fatal. They reflect a serious system failure and should be handled immediately. When confronted with such a message, terminate the program by entering n(o). If you cannot determine what caused the problem, contact your local service provider or another qualified person.

CANNOT SEEK: BLK *disk-block-number* (CONTINUE)

Solaris 10:

CANNOT SEEK: BLK *block-number* (CONTINUE)

Cause

A request to move to the specified block number, *disk-block-number*, in the file system failed. This message indicates a serious problem, probably a hardware failure.

Solaris 10: A request to move to the specified block number, *block-number*, in the file system failed. This message indicates a serious problem, probably a hardware failure.

If you want to continue the file system check, fsck will retry the move and display a list of sector numbers that could not be moved. If the block was part of the virtual memory buffer cache, fsck will terminate with a fatal I/O error message.

Action

If the disk is experiencing hardware problems, the problem will persist. Run `fsck` again to recheck the file system.

If the recheck fails, contact your local service provider or another qualified person.

`CANNOT READ: DISK BLOCK` *disk-block-number*`: I/O ERROR`
` CONTINUE?`

Solaris 10:

`CANNOT READ: DISK BLOCK` *block-number*`: I/O ERROR`
` CONTINUE?`

Cause

A request to read the specified block number, *disk-block-number*, in the file system failed. The message indicates a serious problem, probably a hardware failure.

Solaris 10: A request to read a specified block number, *block-number*, in the file system failed. The message indicates a serious problem, probably a hardware failure.

If you want to continue the file system check, `fsck` will retry the read and display a list of sector numbers that could not be read. If the block was part of the virtual memory buffer cache, `fsck` will terminate with a fatal I/O error message. If `fsck` tries to write back one of the blocks on which the read failed, it will display the following message:

`WRITING ZERO'ED BLOCK` *sector-numbers*`TO DISK`

Action

If the disk is experiencing hardware problems, the problem will persist. Run `fsck` again to recheck the file system. If the recheck fails, contact your local service provider or another qualified person.

`CANNOT WRITE: BLK` *disk-block-number* `(CONTINUE)`

Solaris 10:

`CANNOT WRITE: BLK` *block-number* `(CONTINUE)`

Cause

A request to write the specified block number, *disk-block-number*, in the file system failed.

If you continue the file system check, `fsck` will retry the write and display a list of sector numbers that could not be written. If the block was part of the virtual memory buffer cache, `fsck` will terminate with a fatal I/O error message.

Solaris 10: A request to write a specified block number, *block-number*, in the file system failed.

If you continue the file system check, `fsck` will retry the write and display a list of sector numbers that could not be written. If the block was part of the virtual memory buffer cache, `fsck` will terminate with a fatal I/O error message.

Action

The disk might be write-protected. Check the write-protect lock on the drive. If the disk has hardware problems, the problem will persist. Run `fsck` again to recheck the file system. If the write-protect is not the problem or the recheck fails, contact your local service provider or another qualified person.

Initialization Phase `fsck` Messages

In the initialization phase, command-line syntax is checked. Before the file system check can be performed, `fsck` sets up tables and opens files.

The messages in this section relate to error conditions resulting from command-line options, memory requests, the opening of files, the status of files, file system size checks, and the creation of the scratch file. All such initialization errors terminate `fsck` when it is preening the file system.

`Can’t roll the log for` *device-name*`.`

```
DISCARDING THE LOG MAY DISCARD PENDING TRANSACTIONS.
DISCARD THE LOG AND CONTINUE?
```

Cause

`fsck` was unable to flush the transaction log of a logging UFS file system prior to checking the file system for errors.

Action

Answering yes means the file system operations that were in the log, but had not been applied to the file system, are lost. In this case, `fsck` runs the same checks it always runs and asks the following question in phase 5:

```
FREE BLK COUNT(S) WRONG IN SUPERBLK (SALVAGE)
```

Answering yes at this point reclaims the blocks that were used for the log. The next time the file system is mounted with logging enabled, the log will be recreated.

Answering no preserves the log and exits, but the file system isn't mountable.

`bad inode number` *inode-number* `to ginode`

Cause

An internal error occurred because of a nonexistent inode *inode-number*. `fsck` exits.

Action

Contact your local service provider or another qualified person.

`cannot alloc` *size-of-block map* `bytes for blockmap`
`cannot alloc` *size-of-free map* `bytes for freemap`
`cannot alloc` *size-of-state map* `bytes for statemap`
`cannot alloc` *size-of-lncntp* `bytes for lncntp`

Cause

Request for memory for its internal tables failed. `fsck` terminates. This message indicates a serious system failure that should be handled immediately. This condition might occur if other processes are using a very large amount of system resources.

Action

Killing other processes might solve the problem. If not, contact your local service provider or another qualified person.

`Can’t open checklist file:` *filename*

Cause

The file system checklist file *filename* (usually `/etc/vfstab`) cannot be opened for reading. `fsck` terminates.

Action

Check if the file exists and if its access modes permit read access.

`Can’t open` *filename*

Cause

`fsck` cannot open file system *filename*. When running interactively, `fsck` ignores this file system and continues checking the next file system given.

Action

Check to see if read and write access to the raw device file for the file system is permitted.

`Can’t stat root`

Cause

`fsck` request for statistics about the root directory failed. `fsck` terminates.

Action

This message indicates a serious system failure. Contact your local service provider or another qualified person.

`Can’t stat` *filename*
`Can’t make sense out of name` *filename*

Cause

`fsck` request for statistics about the file system *filename* failed. When running interactively, `fsck` ignores this file system and continues checking the next file system given.

Action

Check if the file system exists and check its access modes.

filename: (NO WRITE)

Cause

Either the -n option was specified or fsck could not open the file system *filename* for writing. When fsck is running in no-write mode, all diagnostic messages are displayed, but fsck does not attempt to fix anything.

Action

If -n was not specified, check the type of the file specified. It might be the name of a regular file.

IMPOSSIBLE MINFREE=*percent* IN SUPERBLOCK (SET TO DEFAULT)

Cause

The superblock minimum space percentage is greater than 99 percent or less than 0 percent.

Action

To set the minfree parameter to the default 10 percent, type y at the default prompt. To ignore the error condition, type n at the default prompt.

filename: BAD SUPER BLOCK: *message*

```
USE AN ALTERNATE SUPER-BLOCK TO SUPPLY NEEDED INFORMATION;
e.g., fsck[-f ufs] -o b=# [special ...]
where # is the alternate superblock.  See fsck_ufs(1M)
```

Cause

The superblock has been corrupted.

Action

One of the following messages might be displayed:

```
CPG OUT OF RANGE
FRAGS PER BLOCK OR FRAGSIZE WRONG
INODES PER GROUP OUT OF RANGE
INOPB NONSENSICAL RELATIVE TO BSIZE
MAGIC NUMBER WRONG
NCG OUT OF RANGE
NCYL IS INCONSISTENT WITH NCG*CPG
NUMBER OF DATA BLOCKS OUT OF RANGE
NUMBER OF DIRECTORIES OUT OF RANGE
ROTATIONAL POSITION TABLE SIZE OUT OF RANGE
SIZE OF CYLINDER GROUP SUMMARY AREA WRONG
SIZE TOO LARGE
BAD VALUES IN SUPERBLOCK
```

Try to rerun fsck with an alternative superblock. Specifying block 32 is a good first choice. You can locate an alternative copy of the superblock by running the newfs -N command on the slice. Be sure to specify the -N option; otherwise, newfs overwrites the existing file system.

`UNDEFINED OPTIMIZATION IN SUPERBLOCK (SET TO DEFAULT)`

Cause

The superblock optimization parameter is neither `OPT_TIME` nor `OPT_SPACE`.

Action

To minimize the time to perform operations on the file system, type `y` at the `SET TO DEFAULT` prompt. To ignore this error condition, type `n`.

Phase 1: Check Blocks and Sizes Messages

This phase checks the inode list. It reports error conditions encountered while:

- Checking inode types
- Setting up the zero-link-count table
- Examining inode block numbers for bad or duplicate blocks
- Checking inode size
- Checking inode format

All errors in this phase except `INCORRECT BLOCK COUNT`, `PARTIALLY TRUNCATED INODE`, `PARTIALLY ALLOCATED INODE`, and `UNKNOWN FILE TYPE` terminate `fsck` when it is preening a file system.

These messages (in alphabetical order) might occur in phase 1:

block-number `BAD I=`*inode-number*

Cause

Inode *inode-number* contains a block number *block-number* with a number lower than the number of the first data block in the file system or greater than the number of the last block in the file system. This error condition might generate the `EXCESSIVE BAD BLKS` error message in phase 1 if inode *inode-number* has too many block numbers outside the file system range. This error condition generates the `BAD/DUP` error message in phases 2 and 4.

Action

N/A

`BAD MODE: MAKE IT A FILE?`

Cause

The status of a given inode is set to all 1s, indicating file system damage. This message does not indicate physical disk damage, unless it is displayed repeatedly after `fsck -y` has been run.

Action

Type `y` to reinitialize the inode to a reasonable value.

`BAD STATE` *state-number* `TO BLKERR`

Cause

An internal error has scrambled the `fsck` state map so that it shows the impossible value *state-number*. `fsck` exits immediately.

Action

Contact your local service provider or another qualified person.

fragment-number `DUP I=`*inode-number*

Solaris 10:

block-number `DUP I=`*inode-number*

Cause

Inode *inode-number* contains a block number *fragment-number*, which is already claimed by the same or another inode. This error condition might generate the `EXCESSIVE DUP BLKS` error message in phase 1 if inode *inode-number* has too many block numbers claimed by the same or another inode. This error condition invokes phase 1B and generates the `BAD/DUP` error messages in phases 2 and 4.

Solaris 10: Inode *inode-number* contains a block number *block-number*, which is already claimed by the same or another inode. This error condition might generate the `EXCESSIVE DUP BLKS` error message in phase 1 if inode *inode-number* has too many block numbers claimed by the same or another inode. This error condition invokes phase 1B and generates the `BAD/DUP` error messages in phases 2 and 4.

Action

N/A

`DUP TABLE OVERFLOW (CONTINUE)`

Cause

`fsck` could not allocate memory to track duplicate fragments. If the `-o p` option is specified, the program terminates.

Solaris 10: There is no more room in an internal table in `fsck` containing duplicate block numbers. If the `-o p` option is specified, the program terminates.

Action

To continue the program, type `y` at the `CONTINUE` prompt. When this error occurs, a complete check of the file system is not possible. If another duplicate fragment is found, this error condition repeats. Increase the amount of virtual memory available (by killing some processes, increasing swap space) and run `fsck` again to recheck the file system. To terminate the program, type `n`.

Solaris 10: To continue the program, type `y` at the `CONTINUE` prompt. When this error occurs, a complete check of the file system is not possible. If another duplicate block is found, this error condition repeats. Increase the amount of virtual memory available (by killing some processes, increasing swap space) and run `fsck` again to recheck the file system. To terminate the program, type `n`.

`EXCESSIVE BAD FRAGMENTS I=inode-number (CONTINUE)`

Solaris 10:

`EXCESSIVE BAD BLOCKS I=inode-number (CONTINUE)`

Cause

Too many (usually more than 10) fragments indicate an invalid disk address. If the `-o p` (preen) option is specified, the program terminates.

Solaris 10: Too many (usually more than 10) blocks have a number lower than the number of the first data block in the file system or greater than the number of the last block in the file system associated with inode *inode-number*. If the `-o p` (preen) option is specified, the program terminates.

Action

To continue the program, type `y` at the `CONTINUE` prompt. When this error occurs, a complete check of the file system is not possible. You should run `fsck` again to recheck the file system. To terminate the program, type `n`.

`EXCESSIVE DUP BLKSDUPLICATE FRAGMENTS I=inode-number (CONTINUE)`

Solaris 10:

`EXCESSIVE DUP BLKS I=inode-number (CONTINUE)`

Cause

Too many (usually more than 10) fragments are claimed by the same or another inode or by a free-list. If the `-o p` option is specified, the program terminates.

Solaris 10: Too many (usually more than 10) blocks are claimed by the same or another inode or by a free-list. If the `-o p` option is specified, the program terminates.

Action

To continue the program, type `y` at the `CONTINUE` prompt. When this error occurs, a complete check of the file system is not possible. You should run `fsck` again to recheck the file system. To terminate the program, type `n`.

`INCORRECT DISK BLOCK COUNT I=inode-number (`*number-of-BAD-DUP-or-missing-blocks*
`should be` *number-of-blocks-in-filesystem*`) (CORRECT)`

Solaris 10:

`INCORRECT BLOCK COUNT I=inode-number (`*number-of-BAD-DUP-or-missing-blocks*
`should be` *number-of-blocks-in-filesystem*`) (CORRECT)`

Cause

The disk block count for inode *inode-number* is incorrect.. When preening, `fsck` corrects the count.

Solaris 10: The block count for inode *inode-number* is *number-of-BAD-DUP-or-missing-blocks*, but should be *number-of-blocks-in-filesystem*. When preening, `fsck` corrects the count.

Action

To correct the disk block count of inode *inode-number* by *number-of-blocks-in-file*, type `y` at the `CORRECT` prompt. .

Solaris 10: To replace the block count of inode *inode-number* by *number-of-blocks-in-filesystem*, type `y` at the `CORRECT` prompt. To terminate the program, type `n`.

`LINK COUNT TABLE OVERFLOW (CONTINUE)`

Cause

There is no more room in an internal table for `fsck` containing allocated inodes with a link count of zero. If the `-o p` (preen) option is specified, the program exits and `fsck` has to be completed manually.

Action

To continue the program, type `y` at the `CONTINUE` prompt. If another allocated inode with a zero-link count is found, this error condition repeats. When this error occurs, a complete check of the file system is not possible. You should run `fsck` again to recheck the file system. Increase the virtual memory available by killing some processes or increasing swap space, then run `fsck` again. To terminate the program, type `n`.

`PARTIALLY ALLOCATED INODE I=`*inode-number* `(CLEAR)`

Cause

Inode *inode-number* is neither allocated nor unallocated. If the `-o p` (preen) option is specified, the inode is cleared.

Action

To deallocate the inode *inode-number* by zeroing out its contents, type `y`. This might generate the `UNALLOCATED` error condition in phase 2 for each directory entry pointing to this inode. To ignore the error condition, type `n`. A no response is appropriate only if you intend to take other measures to fix the problem.

`PARTIALLY TRUNCATED INODE I=`*inode-number* `(SALVAGE)`

Cause

`fsck` has found inode *inode-number* whose size is shorter than the number of fragments allocated to it. This condition occurs only if the system crashes while truncating a file. When preening the file system, `fsck` completes the truncation to the specified size.

Solaris 10: `fsck` has found inode *inode-number* whose size is shorter than the number of blocks allocated to it. This condition occurs only if the system crashes while truncating a file. When preening the file system, `fsck` completes the truncation to the specified size.

Action

To complete the truncation to the size specified in the inode, type `y` at the `SALVAGE` prompt. To ignore this error condition, type `n`.

`UNKNOWN FILE TYPE I=`*inode-number* `(CLEAR)`

Cause

The mode word of the inode *inode-number* shows that the inode is not a pipe, character device, block device, regular file, symbolic link, FIFO file, or directory inode. If the `-o p` option is specified, the inode is cleared.

Solaris 10: The mode word of the inode *inode-number* shows that the inode is not a pipe, special character inode, special block inode, regular inode, symbolic link, FIFO file, or directory inode. If the `-o p` option is specified, the inode is cleared.

Action

To deallocate the inode *inode-number* by zeroing its contents, which results in the `UNALLOCATED` error condition in phase 2 for each directory entry pointing to this inode, type `y` at the `CLEAR` prompt. To ignore this error condition, type `n`.

Solaris 10: Phase 1B: Rescan for More DUPS Messages

This sections contains phase 1B `fsck` messages in the current Solaris release.

When a duplicate fragment is found in the file system, this message is displayed:

`fragment DUP I=`*inode-number*

Cause

Inode *inode-number* contains a fragment number *fragment-number* that is already claimed by the same or another inode. This error condition generates the `BAD/DUP` error message in phase 2. Inodes that have overlapping fragments might be determined by examining this error condition and the `DUP` error condition in phase 1. This is simplified by the duplicate fragment report produced at the fsck run.

Action

When a duplicate block is found, the file system is rescanned to find the inode that previously claimed that block.

Phase 1B: Rescan for More DUPS Messages

This section contains fsck messages that are displayed in the Solaris 10 release.

When a duplicate block is found in the file system, this message is displayed:

block-number DUP I=*inode-number*

Cause

Inode *inode-number* contains a block number *block-number* that is already claimed by the same or another inode. This error condition generates the BAD/DUP error message in phase 2. Inodes that have overlapping blocks might be determined by examining this error condition and the DUP error condition in phase 1.

Action

When a duplicate block is found, the file system is rescanned to find the inode that previously claimed that block.

Phase 2: Check Path Names Messages

This phase removes directory entries pointing to bad inodes found in phases 1 and 1B. It reports error conditions resulting from:

- Incorrect root inode mode and status
- Directory inode pointers out of range
- Directory entries pointing to bad inodes
- Directory integrity checks

When the file system is being preened (-o -poption), all errors in this phase terminate fsck, except those related to directories not being a multiple of the block size, duplicate and bad blocks, inodes out of range, and extraneous hard links.

These messages (in alphabetical order) might occur in phase 2:

BAD INODE *state-number* TO DESCEND

Cause

An fsck internal error has passed an invalid state *state-number* to the routine that descends the file system directory structure. fsck exits.

Action

If this error message is displayed, contact your local service provider or another qualified person.

BAD INODE NUMBER FOR '.' I=*inode-number* OWNER=*UID* MODE=*file-mode*
SIZE=*file-size* MTIME=*modification-time* DIR=*filename* (FIX)

Cause

A directory *inode-number* has been found whose inode number for "." does not equal *inode-number*.

Action

To change the inode number for "." to be equal to *inode-number*, type `y` at the `FIX` prompt To leave the inode numbers for "." unchanged, type `n`.

`BAD INODE NUMBER FOR '..' I=`*inode-number* `OWNER=`*UID* `MODE=`*file-mode* `SIZE=`*file-size* `MTIME=`*modification-time* `DIR=`*filename* `(FIX)`

Cause

A directory *inode-number* has been found whose inode number for ".." does not equal the parent of *inode-number*.

Action

To change the inode number for ".." to be equal to the parent of *inode-number*, type `y` at the `FIX` prompt. (Note that ".." in the root inode points to itself.) To leave the inode number for ".." unchanged, type `n`.

`BAD RETURN STATE` *state-number* `FROM DESCEND`

Cause

An `fsck` internal error has returned an impossible state *state-number* from the routine that descends the file system directory structure. `fsck` exits.

Action

If this message is displayed, contact your local service provider or another qualified person.

`BAD STATE` *state-number* `FOR ROOT INODE`

Cause

An internal error has assigned an impossible state *state-number* to the root inode. `fsck` exits.

Action

If this error message is displayed, contact your local service provider or another qualified person.

`BAD STATE` *state-number* `FOR INODE=`*inode-number*

Cause

An internal error has assigned an impossible state *state-number* to inode *inode-number*. `fsck` exits.

Action

If this error message is displayed, contact your local service provider or another qualified person.

`DIRECTORY TOO SHORT I=`*inode-number* `OWNER=`*UID* `MODE=`*file-mode* `SIZE=`*file-size* `MTIME=`*modification-time* `DIR=`*filename* `(FIX)`

Cause

A directory *filename* has been found whose size *file-size* is less than the minimum directory size. The owner *UID*, mode *file-mode*, size *file-size*, modify time *modification-time*, and directory name *filename* are displayed.

Action

To increase the size of the directory to the minimum directory size, type `y` at the `FIX` prompt. To ignore this directory, type `n`.

`DIRECTORY` *filename*`: LENGTH` *file-size* `NOT MULTIPLE OF` *disk-block-size* `(ADJUST)`

Solaris 10:

`DIRECTORY` *filename*`: LENGTH` *file-size* `NOT MULTIPLE OF` *block-number* `(ADJUST)`

Cause

A directory *filename* has been found with size *file-size* that is not a multiple of the directory block size *disk-block-size*.

Solaris 10:

A directory *filename* has been found with size *file-size* that is not a multiple of the directory block size *block-number*.

Action

To round up the length to the appropriate disk block size, type `y`. When preening the file system (`-o p` option), `fsck` only displays a warning and adjusts the directory. To ignore this condition, type `n`.

Solaris 10:

To round up the length to the appropriate block size, type `y`. When preening the file system (`-o p` option), `fsck` only displays a warning and adjusts the directory. To ignore this condition, type `n`.

`DIRECTORY CORRUPTED I=`*inode-number* `OWNER=`*UID* `MODE=`*file-mode*
`SIZE=`*file-size* `MTIME=`*modification-time* `DIR=`*filename* `(SALVAGE)`

Cause

A directory with an inconsistent internal state has been found.

Action

To throw away all entries up to the next directory boundary (usually a 512-byte boundary), type `y` at the `SALVAGE` prompt. This drastic action can throw away up to 42 entries. Take this action only after other recovery efforts have failed. To skip to the next directory boundary and resume reading, but not modify the directory, type `n`.

`DUP/BAD I=`*inode-number* `OWNER=O MODE=M SIZE=`*file-size*
`MTIME=`*modification-time* `TYPE=`*filename* `(REMOVE)`

Cause

Phase 1 or phase 1B found duplicate fragments or bad fragments associated with directory or file entry *filename*, inode *inode-number*. The owner *UID*, mode *file-mode*, size *file-size*, modification time *modification-time*, and directory or file name *filename* are displayed. If the `-op` (preen) option is specified, the duplicate/bad fragments are removed.

Solaris 10:

Phase 1 or phase 1B found duplicate blocks or bad blocks associated with directory or file entry *filename*, inode *inode-number*. The owner *UID*, mode *file-mode*, size *file-size*, modification time *modification-time*, and directory or file name *filename* are displayed. If the `-op` (preen) option is specified, the duplicate/bad blocks are removed.

Action

To remove the directory or file entry *filename*, type `y` at the `REMOVE` prompt. To ignore this error condition, type `n`.

`DUPS/BAD IN ROOT INODE (REALLOCATE)`

Cause

Phase 1 or phase 1B has found duplicate fragments or bad fragments in the root inode, (inode number 20, of the file system.

Solaris 10:

Phase 1 or phase 1B has found duplicate blocks or bad blocks in the root inode (usually inode number 2 of the file system.

Action

To clear the existing contents of the root inode and reallocate it, type `y` at the `REALLOCATE` prompt. The files and directories usually found in the root inode will be recovered in phase 3 and put into the `lost+found` directory. If the attempt to allocate the root fails, `fsck` will exit with: `CANNOT ALLOCATE ROOT INODE`. Type `n` to get the `CONTINUE` prompt. Type: `y` to respond to the `CONTINUE` prompt, and ignore the `DUPS/BAD` error condition in the root inode and continue running the file system check. If the root inode is not correct, this might generate many other error messages. Type `n` to terminate the program.

`EXTRA '.' ENTRY I=`*inode-number* `OWNER=`*UID* `MODE=`*file-mode*
`SIZE=`*file-size* `MTIME=`*modification-time* `DIR=`*filename* `(FIX)`

Cause

A directory *inode-number* has been found that has more than one entry for "`.`".

Action

To remove the extra entry for "`.`" type `y` at the `FIX` prompt. To leave the directory unchanged, type `n`.

`EXTRA '..' ENTRY I=`*inode-number* `OWNER=`*UID* `MODE=`*file-mode*
`SIZE=`*file-size* `MTIME=`*modification-time* `DIR=`*filename* `(FIX)`

Cause

A directory *inode-number* has been found that has more than one entry for "`..`" (the parent directory).

Action

To remove the extra entry for '`..`' (the parent directory), type `y` at the `FIX` prompt. To leave the directory unchanged, type `n`.

hard-link-number `IS AN EXTRANEOUS HARD LINK TO A DIRECTORY` *filename* `(REMOVE)`

Cause

`fsck` has found an extraneous hard link *hard-link-number* to a directory *filename*. When preening (`-o p` option), `fsck` ignores the extraneous hard links.

Action

To delete the extraneous entry *hard-link-number* type `y` at the `REMOVE` prompt. To ignore the error condition, type `n`.

inode-number `OUT OF RANGE I=`*inode-number* `NAME=`*filename* `(REMOVE)`

Cause

A directory entry *filename* has an inode number *inode-number* that is greater than the end of the inode list. If the `-p` (preen) option is specified, the inode will be removed automatically.

Action

To delete the directory entry *filename* type `y` at the `REMOVE` prompt. To ignore the error condition, type `n`.

`MISSING '.' I=`*inode-number* `OWNER=`*UID* `MODE=`*file-mode* `SIZE=`*file-size*
`MTIME=`*modification-time* `DIR=`*filename* `(FIX)`

Cause

A directory *inode-number* has been found whose first entry (the entry for "`.`") is unallocated.

Action

To build an entry for "`.`" with inode number equal to *inode-number*, type `y` at the `FIX` prompt. To leave the directory unchanged, type `n`.

`MISSING '.' I=`*inode-number* `OWNER=`*UID* `MODE=`*file-mode* `SIZE=`*file-size*
`MTIME=`*modification-time* `DIR=`*filename* `CANNOT FIX, FIRST ENTRY IN`
`DIRECTORY CONTAINS` *filename*

Cause

A directory *inode-number* has been found whose first entry is *filename*. `fsck` cannot resolve this problem.

Action

If this error message is displayed, contact your local service provider or another qualified person.

MISSING '.' I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename* CANNOT FIX, INSUFFICIENT SPACE TO ADD '.'

Cause

A directory *inode-number* has been found whose first entry is not "`.`". `fsck` cannot resolve the problem.

Action

If this error message is displayed, contact your local service provider or another qualified person.

MISSING '..' I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename* (FIX)

Cause

A directory *inode-number* has been found whose second entry is unallocated.

Action

To build an entry for "`..`" with inode number equal to the parent of *inode-number*, type `y` at the `FIX` prompt. (Note that "`..`" in the root inode points to itself.) To leave the directory unchanged, type `n`.

MISSING '..' I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename* CANNOT FIX, SECOND ENTRY IN DIRECTORY CONTAINS *filename*

Cause

A directory *inode-number* has been found whose second entry is *filename*. `fsck` cannot resolve this problem.

Action

If this error message is displayed, contact your local service provider or another qualified person.

MISSING '..' I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time* DIR=*filename* CANNOT FIX, INSUFFICIENT SPACE TO ADD '..'

Cause

A directory *inode-number* has been found whose second entry is not "`..`" (the parent directory). `fsck` cannot resolve this problem.

Action

If this error message is displayed, contact your local service provider or another qualified person.

NAME TOO LONG *filename*

Cause

An excessively long path name has been found, which usually indicates loops in the file system name space. This error can occur if a privileged user has made circular links to directories.

Action

Remove the circular links.

`ROOT INODE UNALLOCATED (ALLOCATE)`

Cause

The root inode (usually inode number 2) has no allocate-mode bits.

Action

To allocate inode 2 as the root inode, type `y` at the `ALLOCATE` prompt. The files and directories usually found in the root inode will be recovered in phase 3 and put into the `lost+found` directory. If the attempt to allocate the root inode fails, `fsck` displays this message and exits: `CANNOT ALLOCATE ROOT INODE`. To terminate the program, type `n`.

`ROOT INODE NOT DIRECTORY (REALLOCATE)`

Cause

The root inode (usually inode number 2) of the file system is not a directory inode.

Action

To clear the existing contents of the root inode and reallocate it, type `y` at the `REALLOCATE` prompt. The files and directories usually found in the root inode will be recovered in phase 3 and put into the `lost+found` directory. If the attempt to allocate the root inode fails, `fsck` displays this message and exits: `CANNOT ALLOCATE ROOT INODE`. To have `fsck` prompt with `FIX`, type `n`.

`UNALLOCATED I=`*inode-number* `OWNER=`*UID* `MODE=`*file-mode* `SIZE=`*file-size*
`MTIME=`*modification-time* `type=`*filename*`(REMOVE)`

Cause

A directory or file entry *filename* points to an unallocated inode *inode-number*. The owner *UID*, mode *file-mode*, size *file-size*, modify time *modification-time*, and file name *filename* are displayed.

Action

To delete the directory entry *filename*, type `y` at the `REMOVE` prompt. To ignore the error condition, type `n`.

`ZERO LENGTH DIRECTORY I=`*inode-number* `OWNER=`*UID* `MODE=`*file-mode*
`SIZE=`*file-size* `MTIME=`*modification-time* `DIR=`*filename* `(REMOVE)`

Cause

A directory entry *filename* has a size *file-size* that is zero. The owner UID, mode *file-mode*, size *file-size*, modify time *modification-time*, and directory name *filename* are displayed.

Action

To remove the directory entry *filename*, type `y` at the `REMOVE` prompt. This results in the `BAD/DUP` error message in phase 4. To ignore the error condition, type `n`.

Phase 3: Check Connectivity Messages

This phase checks the directories examined in phase 2 and reports error conditions resulting from:

- Unreferenced directories
- Missing or full `lost+found` directories

These messages (in alphabetical order) might occur in phase 3:

`BAD INODE` *state-number* `TO DESCEND`

Cause

An internal error has caused an impossible state *state-number* to be passed to the routine that descends the file system directory structure. `fsck` exits.

Action

If this occurs, contact your local service provider or another qualified person.

`DIR I=`*inode-number1* `CONNECTED. PARENT WAS I=`*inode-number2*

Cause

This is an advisory message indicating a directory inode *inode-number1* was successfully connected to the `lost+found` directory. The parent inode *inode-number2* of the directory inode *inode-number1* is replaced by the inode number of the `lost+found` directory.

Action

N/A

`DIRECTORY` *filename* `LENGTH` *file-size* `NOT MULTIPLE OF` *disk-block-size* `(ADJUST)`

Solaris 10:

`DIRECTORY` *filename* `LENGTH` *file-size* `NOT MULTIPLE OF` *block-number* `(ADJUST)`

Cause

A directory *filename* has been found with size *file-size* that is not a multiple of the directory block size B. (This condition can recur in phase 3 if it is not adjusted in phase 2.)

Action

To round up the length to the appropriate disk block size, type `y` at the `ADJUST` prompt. When preening, `fsck` displays a warning and adjusts the directory. To ignore this error condition, type `n`.

Solaris 10:

To round up the length to the appropriate block size, type y at the ADJUST prompt. When preening, fsck displays a warning and adjusts the directory. To ignore this error condition, type n.

```
lost+found IS NOT A DIRECTORY (REALLOCATE)
```

Cause

The entry for lost+found is not a directory.

Action

To allocate a directory inode and change the lost+found directory to reference it, type y at the REALLOCATE prompt. The previous inode reference by the lost+found directory is not cleared and it will either be reclaimed as an unreferenced inode or have its link count adjusted later in this phase. Inability to create a lost+found directory displays the message: SORRY. CANNOT CREATE lost+found DIRECTORY and aborts the attempt to link up the lost inode, which generates the UNREF error message in phase 4. To abort the attempt to link up the lost inode, which generates the UNREF error message in phase 4, type n.

```
NO lost+found DIRECTORY (CREATE)
```

Cause

There is no lost+found directory in the root directory of the file system. When preening, fsck tries to create a lost+found directory.

Action

To create a lost+found directory in the root of the file system, type y at the CREATE prompt. This might lead to the message NO SPACE LEFT IN / (EXPAND). If the lost+found directory cannot be created, fsck displays the message: SORRY. CANNOT CREATE lost+found DIRECTORY and aborts the attempt to link up the lost inode. This in turn generates the UNREF error message later in phase 4. To abort the attempt to link up the lost inode, type n.

```
NO SPACE LEFT IN /lost+found (EXPAND)
```

Cause

Another entry cannot be added to the lost+found directory in the root directory of the file system because no space is available. When preening, fsck expands the lost+found directory.

Action

To expand the lost+found directory to make room for the new entry, type y at the EXPAND prompt. If the attempted expansion fails, fsck displays: SORRY. NO SPACE IN lost+found DIRECTORY and aborts the request to link a file to the lost+found directory. This error generates the UNREF error message later in phase 4. Delete any unnecessary entries in the lost+found directory. This error terminates fsck when preening is in effect. To abort the attempt to link up the lost inode, type n.

UNREF DIR I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size*
MTIME=*modification-time* (RECONNECT)

Cause

The directory inode *inode-number* was not connected to a directory entry when the file system was traversed. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of directory inode *inode-number* are displayed. When preening, `fsck` reconnects the non-empty directory inode if the directory size is non-zero. Otherwise, `fsck` clears the directory inode.

Action

To reconnect the directory inode *inode-number* into the `lost+found` directory, type `y` at the `RECONNECT` prompt. If the directory is successfully reconnected, a `CONNECTED` message is displayed. Otherwise, one of the `lost+found` error messages is displayed. To ignore this error condition, type `n`. This error causes the `UNREF` error condition in phase 4.

Phase 4: Check Reference Counts Messages

This phase checks the link count information obtained in phases 2 and 3. It reports error conditions resulting from:

- Unreferenced files
- A missing or full `lost+found` directory
- Incorrect link counts for files, directories, symbolic links, or special files
- Unreferenced files, symbolic links, and directories
- Bad or duplicate fragments in files and directories

  Solaris 10:

  Bad or duplicate blocks in files and directories
- Incorrect total free-inode counts

All errors in this phase (except running out of space in the `lost+found` directory) are correctable when the file system is being preened.

These messages (in alphabetical order) might occur in phase 4:

`BAD/DUP` *type* `I=`*inode-number* `OWNER=`*UID* `MODE=`*file-mode* `SIZE=`*file-size*
`MTIME=`*modification-time* `(CLEAR)`

Cause

Phase 1 or phase 1B found duplicate fragments or bad fragments associated with file or directory inode *inode-number*. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of inode *inode-number* are displayed.

Solaris 10:

Phase 1 or phase 1B found duplicate blocks or bad blocks associated with file or directory inode *inode-number*. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of inode *inode-number* are displayed.

Action

To deallocate inode *inode-number* by zeroing its contents, type y at the CLEAR prompt. To ignore this error condition, type n.

```
(CLEAR)
```

Cause

The inode mentioned in the UNREF error message immediately preceding cannot be reconnected. This message does not display if the file system is being preened because lack of space to reconnect files terminates fsck.

Action

To deallocate the inode by zeroing out its contents, type y at the CLEAR prompt. To ignore the preceding error condition, type n.

LINK COUNT *type* I=*inode-number* OWNER=*UID* MODE=*file-mode*
SIZE=*file-size*
MTIME=*modification-time* COUNT *link-count* SHOULD BE
corrected-link-count (ADJUST)

Cause

The link count for directory or file inode *inode-number* is *link-count* but should be *corrected-link-count*. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of inode *inode-number* are displayed. If the -o p option is specified, the link count is adjusted unless the number of references is increasing. This condition does not occur unless there is a hardware failure. When the number of references is increasing during preening, fsck displays this message and exits: LINK COUNT INCREASING

Action

To replace the link count of directory or file inode *inode-number* with *corrected-link-count*, type y at the ADJUST prompt. To ignore this error condition, type n.

```
lost+found IS NOT A DIRECTORY (REALLOCATE)
```

Cause

The entry for lost+found is not a directory.

Action

To allocate a directory inode and change the lost+found directory to reference it, type y at the REALLOCATE prompt. The previous inode reference by the lost+found directory is not cleared. It will either be reclaimed as an unreferenced inode or have its link count adjusted later in this phase. Inability to create a lost+found directory displays this message: SORRY. CANNOT CREATE lost+found DIRECTORY and aborts the attempt to link up the lost inode. This error generates the UNREF error message later in phase 4. To abort the attempt to link up the lost inode, type n.

```
NO lost+found DIRECTORY (CREATE)
```

Cause

There is no lost+found directory in the root directory of the file system. When preening, fsck tries to create a lost+found directory.

Action

To create a lost+found directory in the root of the file system, type y at the CREATE prompt. If the lost+found directory cannot be created, fsck displays the message: SORRY. CANNOT CREATE lost+found DIRECTORY and aborts the attempt to link up the lost inode. This error in turn generates the UNREF error message later in phase 4. To abort the attempt to link up the lost inode, type n.

NO SPACE LEFT IN / lost+found (EXPAND)

Cause

There is no space to add another entry to the lost+found directory in the root directory of the file system. When preening, fsck expands the lost+found directory.

Action

To expand the lost+found directory to make room for the new entry, type y at the EXPAND prompt. If the attempted expansion fails, fsck displays the message: SORRY. NO SPACE IN lost+found DIRECTORY and aborts the request to link a file to the lost+found directory. This error generates the UNREF error message later in phase 4. Delete any unnecessary entries in the lost+found directory. This error terminates fsck when preening (-o p option) is in effect. To abort the attempt to link up the lost inode, type n.

UNREF FILE I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size*
MTIME=*modification-time* (RECONNECT)

Cause

File inode *inode-number* was not connected to a directory entry when the file system was traversed. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of inode *inode-number* are displayed. When fsck is preening, the file is cleared if either its size or its link count is zero; otherwise, it is reconnected.

Action

To reconnect inode *inode-number* to the file system in the lost+found directory, type y. This error might generate the lost+found error message in phase 4 if there are problems connecting inode *inode-number* to the lost+found directory. To ignore this error condition, type n. This error always invokes the CLEAR error condition in phase 4.

UNREF *type* I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size*
MTIME=*modification-time* (CLEAR)

Cause

Inode *inode-number* (whose *type* is directory or file) was not connected to a directory entry when the file system was traversed. The owner *UID*, mode *file-mode*, size *file-size*, and modification time *modification-time* of inode *inode-number* are displayed. When fsck is preening, the file is cleared if either its size or its link count is zero; otherwise, it is reconnected.

Action
: To deallocate inode *inode-number* by zeroing its contents, type y at the CLEAR prompt. To ignore this error condition, type n.

ZERO LENGTH DIRECTORY I=*inode-number* OWNER=*UID* MODE=*file-mode* SIZE=*file-size* MTIME=*modification-time*(CLEAR)

Cause
: A directory entry *filename* has a size *file-size* that is zero. The owner *UID*, mode *file-mode*, size *file-size*, modification time *modification-time*, and directory name *filename* are displayed.

Action
: To deallocate the directory inode *inode-number* by zeroing out its contents, type y. To ignore the error condition, type n.

Phase 5: Check Cylinder Groups Messages

This section contains phase 5 fsck messages in the current Solaris release.

This phase checks the free-fragment and used-inode maps. It reports error conditions resulting from:

- Allocated inodes missing from used-inode maps
- Free fragments missing from free-fragment maps
- Free inodes in the used-inode maps
- Incorrect total free-fragment count
- Incorrect total used inode count

These messages (in alphabetical order) might occur in phase 5:

FRAG BITMAP WRONG (CORRECTED)

Cause
: A cylinder group fragment map is missing some free fragments. During preening, fsck reconstructs the maps.

Action
: To reconstruct the free-fragment map, type y at the SALVAGE prompt. To ignore this error condition, type n.

CG *cg-number*: BAD MAGIC NUMBER

Cause

The magic number of cylinder group *cg-number* is wrong. This error usually indicates that the cylinder group maps have been destroyed. When running interactively, the cylinder group is marked as needing reconstruction. `fsck` terminates if the file system is being preened.

Action

If this occurs, contact your local service provider or another qualified person.

`CORRECT GLOBAL SUMMARY (SALVAGE)`

Cause

The summary information is incorrect. When preening, `fsck` recomputes the summary information.

Action

To reconstruct the summary information, type `y` at the `SALVAGE` prompt. To ignore this error condition, type `n`.

Phase 5: Check Cylinder Groups Messages

This sections contains phase 5 `fsck` messages in the Solaris 10 initial 3/05 release.

This phase checks the free-block and used-inode maps. It reports error conditions resulting from:

- Allocated inodes missing from used-inode maps
- Free blocks missing from free-block maps
- Free inodes in the used-inode maps
- Incorrect total free-block count
- Incorrect total used inode count

These messages (in alphabetical order) might occur in phase 5:

`BLK(S) MISSING IN BIT MAPS (SALVAGE)`

Cause

A cylinder group block map is missing some free blocks. During preening, `fsck` reconstructs the maps.

Action

To reconstruct the free-block map, type `y` at the `SALVAGE` prompt. To ignore this error condition, type `n`.

`CG` *character-for-command-option*`: BAD MAGIC NUMBER`

Cause

The magic number of cylinder group *character-for-command-option* is wrong. This error usually indicates that the cylinder group maps have been destroyed. When running interactively, the cylinder group is marked as needing reconstruction. `fsck` terminates if the file system is being preened.

Action

If this occurs, contact your local service provider or another qualified person.

```
FREE BLK COUNT(S) WRONG IN SUPERBLK (SALVAGE)
```

Cause

The actual count of free blocks does not match the count of free blocks in the superblock of the file system. If the `-o p` option was specified, the free-block count in the superblock is fixed automatically.

Action

To reconstruct the superblock free-block information, type `y` at the `SALVAGE` prompt. To ignore this error condition, type `n`.

```
SUMMARY INFORMATION BAD (SALVAGE)
```

Cause

The summary information is incorrect. When preening, `fsck` recomputes the summary information.

Action

To reconstruct the summary information, type `y` at the `SALVAGE` prompt. To ignore this error condition, type `n`.

`fsck` Summary Messages

This section contains `fsck` summary messages in the current Solaris release. If you are not running at least the , Solaris 10 6/06 release, these messages are displayed in the cleanup phase. For more information, see “Cleanup Phase Messages” on page 294.

Once a file system has been checked, a few summary messages are displayed.

number-of `files,` *number-of-files*
`used,` *number-of-files* `free (`*number-of* `frags,` *number-of* `blocks,`
percent `fragmentation)`

This message indicates that the file system checked contains *number-of* files using *number-of* fragment-sized blocks, and that there are *number-of* fragment-sized blocks free in the file system. The numbers in parentheses break the free count down into *number-of* free fragments, *number-of* free full-sized blocks, and the *percent* fragmentation.

```
***** FILE SYSTEM WAS MODIFIED *****
```

This message indicates that the file system was modified by `fsck`. There is no need to rerun `fsck` if you see this message. This message is just informational about `fsck`'s corrective actions.

Cleanup Phase Messages

This section contains `fsck` cleanup phase messages in the Solaris 10 release. In this Solaris release, similar messages can be found in the `fsck` summary phase. See "`fsck` Summary Messages" on page 293 for more information.

Once a file system has been checked, a few cleanup functions are performed. The cleanup phase displays the following status messages.

number-of `files,` *number-of-files*
`used,` *number-of-files* `free (`*number-of* `frags,` *number-of* `blocks,`
percent `fragmentation)`

This message indicates that the file system checked contains *number-of* files using *number-of* fragment-sized blocks, and that there are *number-of* fragment-sized blocks free in the file system. The numbers in parentheses break the free count down into *number-of* free fragments, *number-of* free full-sized blocks, and the *percent* fragmentation.

```
***** FILE SYSTEM WAS MODIFIED *****
```

This message indicates that the file system was modified by `fsck`. If this file system is mounted or is the current root (`/`) file system, reboot. If the file system is mounted, you might need to unmount it and run `fsck` again; otherwise, the work done by `fsck` might be undone by the in-core copies of tables.

filename `FILE SYSTEM STATE SET TO OKAY`

This message indicates that file system *filename* was marked as stable. Use the `fsck -m` command to determine if the file system needs checking.

```
filename FILE SYSTEM STATE NOT SET TO OKAY
```

This message indicates that file system filename was not marked as stable. Use the `fsck -m` command to determine if the file system needs checking.

CHAPTER 21

Troubleshooting Software Package Problems (Tasks)

This chapter describes problems you might encounter when installing or removing software packages. The Specific Software Package Installation Errors section describes package installation and administration errors you might encounter. The General Software Package Installation Problems section describes behavioral problems that might not display an error message.

This is a list of information in this chapter:

For information about managing software packages, see Chapter 19, "Managing Software (Overview)," in *System Administration Guide: Basic Administration*.

Troubleshooting Software Package Symbolic Link Problems

In previous Solaris releases, there was no way to specify a symbolic link target in the `pkgmap` file when creating a software package. This meant a package or patch-related symbolic link was always followed to the source of the symbolic link rather than to the target of the symbolic link when a package was added with the `pkgadd` command. This created problems when upgrading a package or a patch package that needed to change a symbolic link target destination to something else.

Now, the default behavior is that if a package needs to change the target of a symbolic link to something else, the target of the symbolic link and not the source of the symbolic link is inspected by the `pkgadd` command.

Unfortunately, this means that some packages may or may not conform to the new `pkgadd` behavior.

The PKG_NONABI_SYMLINKS environment variable might help you transition between the old and new pkgadd symbolic link behaviors. If this environment variable is set to true, pkgadd follows the source of the symbolic link.

Setting this variable enables a non-conforming package to revert to the old behavior if set by the administrator before adding a package with the pkgadd command.

The new pkgadd symbolic link behavior might cause an existing package to fail when added with the pkgadd command. You might see the following error message in this situation:

```
unable to create symbolic link to <path>
```

If a package doesn't install due to this problem, do the following:

1. If this is a Sun-supplied package, call the Resolution Center and report the non-conforming package name.
2. Set the PKG_NONABI_SYMLINKS environment variable and try adding the package with the pkgadd command again.

   ```
   # PKG_NONABI_SYMLINKS=true
   # export PKG_NONABI_SYMLINKS
   # pkgadd pkg-name
   ```

Specific Software Package Installation Errors

```
WARNING: filename <not present on Read Only file system>
```

Reason Error Occurred	How to Fix the Problem
This error message indicates that not all of a package's files could be installed. This usually occurs when you are using pkgadd to install a package on a client. In this case, pkgadd attempts to install a package on a file system that is mounted from a server, but pkgadd doesn't have permission to do so.	If you see this warning message during a package installation, you must also install the package on the server. See Chapter 19, "Managing Software (Overview)," in *System Administration Guide: Basic Administration* for details.

General Software Package Installation Problems

Reason Error Occurred	How to Fix the Problem
There is a known problem with adding or removing some packages developed prior to the Solaris 2.5 release and compatible versions. Sometimes, when adding or removing these packages, the installation fails during user interaction or you are prompted for user interaction and your responses are ignored.	Set the following environment variable and try to add the package again. `NONABI_SCRIPTS=TRUE`

Index