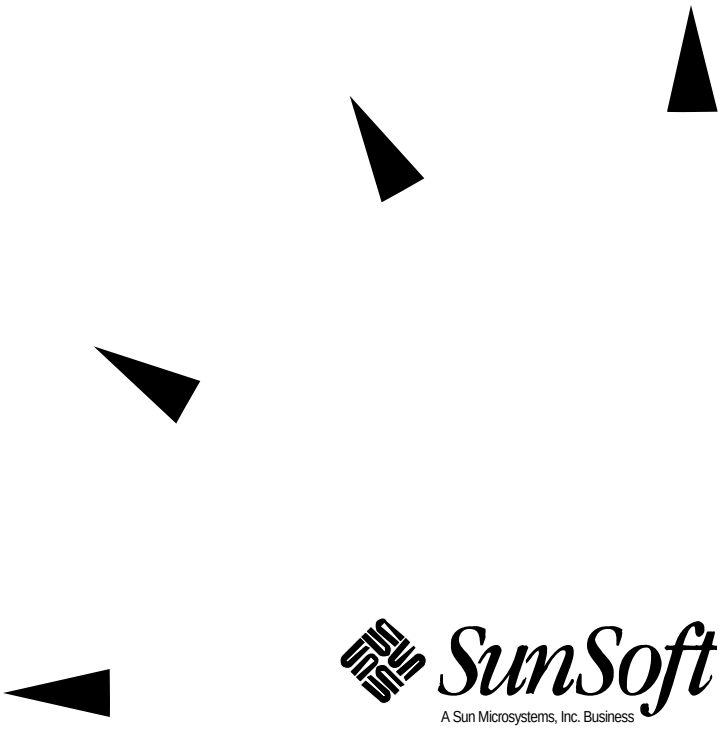
Common Administration Tasks



2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

SunSoft
A Sun Microsystems, Inc. Business

Contents

Tables

Preface

Common Administration Tasks describes some of the administration tasks that are performed on a regular basis, such as halting a system, displaying information about a system, and adding systems to a network.

Who Should Use This Book

This book is written for system administrators who have a working knowledge of the Solaris™ software environment and the SunOS™ system software, and who are familiar with windowing environments and mouse- and menu-driven applications.

Other Books You Need to Use

Several other books help you administer systems:

- For information on adding disks, see *Peripherals Administration*.
- For information on formatting disks, see *Peripherals Administration*.
- For information on setting up printers or mail, or administering users and groups, see *User Accounts, Printers, and Mail Administration*.
- For more information about using Administration Tool, see the *Administration Application Reference Manual*.

- For information on installing unbundled software packages, see *Software and AnswerBook Packages Administration Guide.*
- For information about platform-specific administration see the *Administration Supplement for Solaris Platforms.*

How This Book Is Organized

Common Administration Tasks contains the following chapters:

Chapter 1, "How to Find Information About Solaris Software," describes how to find information on system administration tasks. It also provides a list of books that are in the *Solaris 2.4 System Administrator AnswerBook* and lists the topics covered in each book.

Chapter 2, "Halting a System," describes init states and how to halt a system with the `shutdown` and `init` commands.

Chapter 3, "Boot Files," describes the files used at boot time including, `/etc/system`, `/etc/inittab`, and the run control files.

Chapter 4, "Adding Systems to a Network," provides a description of the different types of system configurations, such as servers and diskless clients, and describes how to use Host Manager to provide support for diskless and dataless clients.

Chapter 5, "Examining and Changing System Information," describes how to find information about a system, such as host name or processor type. It also discusses the format of the `/etc/system` file and how to change the file.

Chapter 6, "Using crontab," describes how to set up a `crontab` file to execute programs.

Chapter 7, "Accessing Remote Files and Systems," describes how to use `rcp`, `rlogin`, `rsh`, and `ftp` to access remote files.

Chapter 8, "Enabling and Using Crash Dumps," describes what a crash dump is, how to enable crash dumps, and what to do with the crash dump after you get one.

What Typographic Changes and Symbols Mean

The following table describes the type changes and symbols used in this book.

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `system% You have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`system% su` `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.
Code samples are included in boxes and may display the following:		
%	C shell prompt	`system%`
#	Superuser prompt, C shell	system#
$	Bourne and Korn shell prompt	$
#	Superuser prompt, Bourne and Korn shells	#

How to Find Information About Solaris Software 1

This chapter contains the following sections:

The first section lists system administration tasks and provides cross references to the appropriate books. The second section lists all of the books in the *Solaris 2.4 System Administrator AnswerBook* and provides a list of the topics covered in each book.

1

Where Is the Information?

Installing Software

Table 1-1 Information on Installing Software

If You Want to...	See This Book:	Which Includes This Related Information:
Install Solaris Software	*Installing Solaris Software* (for your platform)	• Planning installation • Performing the installation • Upgrading
Install or remove a software package	*Software and AnswerBook Packages Administration Guide*	• Installing and removing packages with `pkgadd` and `pkgrm` • Installing and removing packages with Software Manager • Adding client support software with Software Manager
	Administration Application Reference Manual	• Software Manager windows and menus
Install AnswerBook® software	*Software and AnswerBook Packages Administration Guide*	• Hardware and software requirements for AnswerBook installation • Installing AnswerBook software from a local CD-ROM • Installing AnswerBook software over the network from remote CD-ROM • Sharing AnswerBook software over the network • Removing AnswerBook software • AnswerBook administration examples

User Environment Management

Table 1-2 Information on Managing the User Environment

If You Want to...	See This Book:	Which Includes This Related Information:
Add or remove users	*User Accounts, Printers, and Mail Administration*	• Adding user accounts • Removing user accounts • Moving user accounts
	Administration Application Reference Manual	• User Account Manager windows and menus
Set up mail services	*User Accounts, Printers, and Mail Administration*	• Setting up and administering mail and `sendmail` • Customizing `sendmail` configuration files
Give users access to a printer	*User Accounts, Printers, and Mail Administration*	• Setting up printers • Printer administration

File System Management

Table 1-3 Information on File System Management

If You Want to...	See This Book:	Which Includes This Related Information:
Create a file system	*File System Administration*	• Creating a UFS file system • Creating a file system on a diskette
Mount or unmount a local file system	*File System Administration*	• Adding entries to `/etc/vfstab` • Mounting local file system
Mount or unmount a remote file system	*File System Administration*	• Mounting remote file systems
	NFS Administration Guide	• Sharing and unsharing file systems • Setting up automatic sharing of file systems
Format a diskette	*File System Administration*	• Formatting a diskette • Creating a UFS file system on a diskette • Creating a PCFS file system on a diskette
Copy files or file systems	*File System Administration*	• Copying files with `cpio` • Copying files with `tar` • Copying files to diskette or tape
Copy files across a network	*Common Administration Tasks*	• Copying files with `rcp` and `ftp`

Table 1-3 Information on File System Management *(Continued)*

If You Want to...	See This Book:	Which Includes This Related Information:
Back up or restore a file system	*File System Administration*	• Planning a backup schedule • Backing up file systems with `ufsdump` • Restoring file systems with `ufsrestore`
	Networker for Solaris Administrator's Guide	• Configuring Networker for Solaris • Managing backups with Networker for Solaris
Monitor disk usage by users	*File System Administration*	• Using `df` to examine disk usage • Finding large files and directories
Check file system consistency	*File System Administration*	• Using `fsck` to check file systems

Peripheral Installation and Administration

Table 1-4 Information on Installing Peripherals

If You Want to...	See This Book:	Which Includes This Related Information:
Add a printer to a system or network	*User Accounts, Printers, and Mail Administration*	• Setting up printers • Determining printer policies • Managing character sets, filters, fonts, and forms
Add a disk to a system	*Peripherals Administration*	• Preparing disks for use • Adding a system disk • Adding a secondary disk • Disk formatting • Disk slices and labels
Add a tape drive to a system	*Peripherals Administration*	• Adding a SCSI tape drive
Set up terminals or modems	*Peripherals Administration*	• Using Serial Port Manager to connect terminals and modems
	Administration Application Reference Manual	• Using Serial Port Manager • Serial Port Manager Reference
Add or remove device drivers	*Peripherals Administration*	• Adding device drivers • Removing device drivers

Table 1-4 Information on Installing Peripherals *(Continued)*

If You Want to...	See This Book:	Which Includes This Related Information:
Configure Volume Management	*Peripherals Administration*	• Automatically sharing CD-ROMs and diskettes
Add a CD-ROM device	*Peripherals Administration*	• Adding a primary CD-ROM
Format a disk	*Peripherals Administration*	• `format` utility

Network Administration

Table 1-5 Information on Setting up Network Services

If You Want to...	See This Book:	Which Includes This Related Information:
Add a client or standalone system to a network	*Common Administration Tasks*	• Using Host Manager to add clients or standalone systems
Log in to remote machines	*Common Administration Tasks*	• Using `rlogin`
Set up DNS server or client	*Name Services Configuration Guide*	• DNS structure • Setting up clients and servers
Monitor network performance	*Security, Performance, and Accounting Administration*	• Using `ping`, `snoop`, `spray`, `netstat`, `nfsstat`
Configure TCP/IP	*TCP/IP Network Administration Guide*	• Configuring TCP/IP • Configuring routers
Configure PPP	*TCP/IP Network Administration Guide*	• Configuring PPP • Tailoring a PPP link
Configure UUCP	*TCP/IP Network Administration Guide*	• Administering UUCP
Set up an NFS server	*NFS Administration Guide*	• Setting up an NFS server • Troubleshooting
Set up NFS security	*NFS Administration Guide*	• Setting up NFS security
Set up or use autofs	*NFS Administration Guide*	• Setting up `autofs` maps

1

NIS+ Configuration and Administration

Table 1-6 Information on NIS+

If You Want to...	See This Book:	Which Includes This Related Information:
Set up NIS+ root domain	*Name Services Configuration Guide*	• Setting NIS+ root domain
	Name Services Administration Guide	• Creating root master server • Populating tables
Set up NIS+ tables	*Name Services Configuration Guide*	• Populating tables from files or NIS maps
	Name Services Administration Guide	• NIS+ table structure
Set up NIS+ server	*Name Services Configuration Guide*	• Setting up an NIS+ server • NIS+ server security
	Name Services Administration Guide	• NIS+ setup overview • Customizing an NIS+ server
Set up NIS+ client	*Name Services Configuration Guide*	• Setting up NIS+ client
	Name Services Administration Guide	• Initializing an NIS+ client • Initializing client users
Set up NIS+ non-root domain	*Name Services Configuration Guide*	• Setting up an NIS+ non-root domain
	Name Services Administration Guide	• Creating a non-root domain • Populating tables for the non-root domain
Set up Name Service Switch	*Name Services Configuration Guide*	• `nsswitch.conf`, `nsswitch.nisplus`, `nsswitch.nis`, and `nsswitch.files` files
	Name Services Administration Guide	• `nsswitch.conf`, `nsswitch.nisplus`, `nsswitch.nis`, and `nsswitch.files` files
Administer name services with Database Manager	*User Accounts, Printers, and Mail Administration*	• Managing groups with Database Manager
	Administration Application Reference Manual	• Name service management • Database Manager reference
Move to NIS+ from NIS	*NIS+ Transition Guide*	• Designing the NIS+ namespace • NIS+ Security • Using NIS compatibility mode • Prerequisites for transition • Implementing the transition

General System Administration

Table 1-7 Information on General System Administration

If You Want to...	See This Book:	Which Includes This Related Information:
Halt a system	*Common Administration Tasks*	• Shutdown procedures • Init states
Boot a system	*Administration Supplement for Solaris Platforms*	• Boot procedures for specific hardware platforms
	Common Administration Tasks	• Boot files
Examine system information such as host name or IP address	*Common Administration Tasks*	• Using `uname`, `prtconf`, `sysdef` commands
Change system information	*Common Administration Tasks*	• Changing the `/etc/system` file
Start processes that will run later in the day or that will run at regular intervals	*Common Administration Tasks*	• Creating, editing, displaying, and removing `crontab` files • Controlling access to `crontab`
Enable crash dumps	*Common Administration Tasks*	• Enabling crash dumps • Using a crash dump file
Run diagnostic tests on the hardware	*SunDiag User's Guide*	• Scaling SunDiag™ hardware tests • Running SunDiag tests • SunDiag test descriptions • Developing your own tests
Set up auditing	*SunSHIELD Basic Security Module Guide*	• Administering auditing • Audit trail analysis • Audit record descriptions
Manage runtime libraries	*Direct Xlib User's Guide*	• Using Direct XLib software
Control user access to systems	*Security, Performance, and Accounting Administration*	• Restricting access to systems • Passwords • Restricted shells
Control access to files and data	*Security, Performance, and Accounting Administration*	• File permissions • Creating groups • File encryption
Control network access	*Security, Performance, and Accounting Administration*	• Remote login • Secure RPC • Administration Tool Security • Using ASET

Table 1-7 Information on General System Administration *(Continued)*

If You Want to...	See This Book:	Which Includes This Related Information:
Monitor system performance	*Security, Performance, and Accounting Administration*	• Using `vmstat`, `iostat`, and `sar`
Monitor processes	*Security, Performance, and Accounting Administration*	• Changing process priority • Changing process class
Test hardware and software from the PROM	*OpenBoot Command Reference Manual* *OpenBoot Quick Reference*	• Booting and testing your system • Setting NVRAM configuration parameters • Using Forth tools • Loading and executing programs • Debugging • Setting up a TIP connection • Building a bootable diskette

What's In All Those Books?

This section contains a list of the books in the *Solaris 2.4 System Administrator AnswerBook* package and includes a list of the topics covered in each book.

Solaris 1.x to Solaris 2.x Transition Guide

- Overview of Major Changes
- Installation and Configuration Changes
- Using the Compatibility Packages
- Security
- User Environment Administration
- Device Administration
- Startup and Shutdown
- File System Administration
- Setting Up Servers to Support Clients
- Setting Up and Using Printers
- Managing Terminals and Modems
- Network Service Administration
- Using Name Services
- Compilers, Linkers, and Debuggers
- Tools and Resources
- Networking and Internationalization
- System and Device Configuration
- Device Drivers, Streams, and Kernel Debuggers
- Commands Reference Table
- System Calls Reference Table
- Library Routines Reference Table
- System Files Reference Table
- `/` and `/usr` File Systems Changes

x86: Installing Solaris Software

- About Installing Solaris
- Preparing to Install Solaris
- Preparing to Install Solaris Over a Network
- Preparing Custom JumpStart™ Installations
- Using Optional Custom JumpStart Features
- Preparing a System for Upgrade
- Booting and Installing Solaris: Interactive

- Booting and Installing Solaris: Custom JumpStart
- Completing an Upgrade
- Where to Go After Installing Solaris
- Work Sheets for the Solaris Installation Program
- Kernel Architectures
- Sample Custom JumpStart Installation
- Troubleshooting

SPARC: Installing Solaris Software

- About Installing Solaris
- Preparing to Install Solaris
- Preparing to Install Solaris Over a Network
- Preparing Custom JumpStart Installations
- Using Optional Custom JumpStart Features
- Preparing a System for Upgrade
- Booting and Installing Solaris: JumpStart
- Booting and Installing Solaris: Interactive
- Booting and Installing Solaris: Custom JumpStart
- Completing an Upgrade
- Where to Go After Installing Solaris
- Work Sheets for the Solaris Installation Program
- Kernel Architectures
- Sample Custom JumpStart Installation
- Troubleshooting

Administration Application Reference Manual

- Administration Tool Overview
- Name Service Management
- Managing User Accounts
- Managing Printers
- Managing Network Services
- Using Serial Port Manager
- User Account Manager Reference
- Printer Manager Reference
- Host Manager Reference
- Database Manager Reference
- Serial Port Manager Reference

Administration Supplement for Solaris Platforms

- Accessing Devices on a SPARC® System
- Booting a SPARC System
- Setting Up Disks on Your SPARC System
- Managing File Systems on a SPARC System
- Accessing Devices on an x86 System
- Booting an x86 System
- Setting Up Disks on Your x86 System
- Managing File Systems on an x86 System
- Summary of System Administration Differences
- x86: Additional Disk Commands

Software and AnswerBook Packages Administration Guide

- AnswerBook Installation
- AnswerBook Administration
- Installation from a Remote CD Drive
- Concepts of AnswerBook Administration

Common Administration Tasks

- Halting a System
- Booting a System
- Managing Network Services
- Examining and Changing System Information
- Using `crontab`
- Accessing Remote Files and Systems
- Enabling and Using Crash Dumps

Direct Xlib User's Guide

- Installing the Direct Xlib 3.1 Software
- Using the Direct Xlib 3.1 Software

File System Administration

- Understanding and Planning File Systems
- Creating File Systems
- Mounting and Unmounting File Systems

- Copying UFS Files and File Systems
- The Cache File System
- Understanding Backup and Planning a Backup Strategy
- Backing Up Files and File Systems
- Restoring Files and File Systems
- Configuring Additional Swap Space
- Managing Disk Use
- Recognizing File Access Problems
- Checking the Integrity of File Systems
- File System Reference

Name Services Administration Guide

- Setting Up the Root Domain
- Setting Up an NIS+ Client
- Setting Up NIS+ Servers
- Setting Up a Non-Root Domain
- Setting Up NIS+ Tables
- Setting Up the Name Service Switch
- Administering NIS+ Security
- Administering NIS+ Credentials
- Administering NIS+ Access Rights
- Administering NIS+ Groups
- Administering NIS+ Directories
- Administering NIS+ Tables
- Problems and Solutions
- Error Messages
- Information in NIS+ Tables

Name Services Configuration Guide

- Understanding Name Services
- Understanding the NIS+ Namespace
- Understanding NIS+ Tables and Information
- Understanding the Name Service Switch
- Getting Started With NIS+
- Setting Up NIS+
- DNS Structure
- Setting Up DNS Clients
- Setting Up DNS Servers

- Pre-Setup Worksheets

TCP/IP Network Administration Guide

- Introducing TCP/IP Network Administration
- Planning Your Network
- Configuring TCP/IP on the Network
- Configuring Routers
- Troubleshooting TCP/IP
- Understanding PPP
- Preparing Your PPP Configuration
- Configuring PPP
- Maintaining and Troubleshooting PPP
- Tailoring Your PPP Link
- Administering the UUCP System
- Communications Database Files
- NIC Application Form

NFS Administration Guide

- Overview of the Solaris NFS® Environment
- How to Set Up NFS Servers
- How to Use the NFS Environment
- Setting Up and Maintaining NFS Security
- NFS Troubleshooting
- Using Autofs

NIS+ Transition Guide

- Differences Between NIS and NIS+
- Designing the NIS+ Namespace
- Selecting NIS+ Security Measures
- Using NIS Compatibility Mode
- Prerequisites to Transition
- Implementing the Transition

Peripherals Administration

- Terminals and Modems

- Disk Drives
- CD-ROM and Diskette Drives
- Tape Drives
- Device Drivers
- The Service Access Facility
- Connecting Devices to the Serial Port
- `format` Utility

Security, Performance, and Accounting Administration

- Introduction to Security
- Securing System Access
- Securing Files and Data
- Securing the Network
- Monitoring and Controlling Security Using ASET
- Introduction to Performance
- Managing Processes
- Monitoring Performance
- A Guide to Network Performance
- Setting Up and Maintaining Accounting
- Tuning Kernel Parameters
- The Process Scheduler
- Error Messages

SunDiag User's Guide

- Introducing the SunDiag System Exerciser
- The SunDiag OPEN LOOK® Interface
- The SunDiag TTY Interface
- Scaling SunDiag Hardware Tests
- Running Individual SunDiag Tests from the Command Line
- SunDiag Test Descriptions
- Developing Your Own Tests
- Loopback Connectors
- The `what_rev` Utility

SunSHIELD Basic Security Module Guide

- Installation
- Administering Auditing

- Audit Trail Analysis
- Device Allocation
- Audit Record Descriptions
- BSM Man Pages

User Accounts, Printers, and Mail Administration

- Administering User Accounts
- Administering Groups
- Setting Up Printers
- Routine Printer Administration
- Managing Character Sets, Filters, Forms, and Fonts
- Setting Printing Policies
- Troubleshooting Printing Problems
- Understanding Mail Services
- Setting Up and Administering Mail Services
- Setting Up Printer Services Using the Command-Line Interface
- Understanding and Customizing the LP Print Service
- Customizing `sendmail` Configuration Files

OpenBoot Command Reference Manual

- Booting and testing your system
- Setting NVRAM configuration parameters
- Using Forth tools
- Loading and executing programs
- Debugging
- Setting up a TIP connection
- Building a bootable diskette

OpenBoot Quick Reference

- OpenBoot command reference

1

Halting a System 2

This chapter has the following sections:

The SunOS system software is designed to be left running continuously so that the electronic mail and network software can work correctly. You must, however, halt or shut down a system when:

- Turning off system power
- Installing a new release of the operating system
- Anticipating a power outage
- Adding hardware to the system
- Performing maintenance on a file system

This chapter describes system init states (also called run levels) and procedures for shutting down systems. Procedures for booting a system are discussed in the *Administration Supplement for Solaris Platforms.* Boot files are described in Chapter 3, “Boot Files.”

2

Init States

The init state in which the system is running defines what services and resources are available to users. A system can run in only one init state at a time.

The SunOS system software has eight init states, shown in Table 2-1. The default init state, specified in the `/etc/inittab` file, runs at level 3 for SunOS system software.

Table 2-1 System Init States (Run Levels)

Init State/Run Level	Function
0	Power-down state
1	System administrator state (single-user)
2	Multiuser state (resources not exported)
3	Multiuser state (resources exported)
4	Alternative multiuser state (currently unused)
5	Software reboot state
6	Reboot
S,s	Single-user state

The `/sbin/init` program keeps the system running correctly. In addition, `/sbin/init` is the command you use to change init states. You can also specify the init state as an argument to the `shutdown` command with the `-i` option.

There are four types of init states:

- Power-down (run level 0)
- Single-user (run levels 1 and s or S)
- Multiuser (run levels 2 and 3)
- Reboot (run levels 5 and 6)

When preparing to do a system administration task, you need to determine which init state is appropriate for the system and the task at hand.

Table 2-2 describes the use of each init state.

Table 2-2 Uses of Init States

Init State	Run Level	Use This Level...
power-down state	0	To shut down the system so that it is safe to turn off the power.
system administrator state	1	When performing administrative tasks that require you to be the only user on the system. `/` and `/usr` are the only file systems mounted, and you can access only minimum kernel utilities. The terminal from which you issue this command becomes the console. No other users are logged in.
multiuser state	2	For normal operations. Multiple users can access the system and the entire file system. All daemons are running except for NFS server and syslog.
remote resource-sharing state	3	For normal operations with NFS resource-sharing available.
alternative multiuser state	4	This level is currently unavailable.
interactive reboot state	5	When you want to be prompted for a device other than the default boot devices. You can also change to this level by using the `reboot -a` command.
reboot state	6	To shut down the system to run level 0, and then reboot to multiuser level (or whatever level is the default in the `inittab` file).
single-user state	s or S	To run as a single user with all file systems mounted and accessible.

▼ How to Determine a System's Init State

♦ **Type `who -r` and press Return.**
The run level, date and time, process termination status, process ID, and process exit status are displayed.

In this example, `pluto` is at the default multiuser init state (run level 3), the date and time are 3 Feb 6 15:46, the process termination status is 3, the process ID is 0, and process exit status is S:

```
pluto% who -r
   .       run-level 3  Feb  6 15:46     3      0  S
pluto%
```

▼ How to Change the Init State

1. Become superuser.

2. Type `init` *n* and press Return.

To shut down the system:

```
saturn% su
Password:
# init 0
```

To change to single-user state:

```
saturn% su
Password:
# init 1
```

To change to multiuser state, with no NFS server daemons running:

```
saturn% su
Password:
# init 2
```

To change to multiuser state, with NFS server daemons running:

```
saturn% su
Password:
# init 3
```

To shut down and reboot a system:

```
saturn% su
Password:
# init 6
```

Choosing Which Shutdown Command to Use

When preparing to shut down a system, you need to determine which of the following commands is appropriate for the system and the task at hand:

- `/usr/sbin/shutdown`
- `/sbin/init`
- `/usr/sbin/halt`
- `/usr/sbin/reboot`

These commands initiate shutdown procedures, kill all running processes, write data to disk, and shut down the system software to the appropriate run level.

Table 2-3 describes each of the shutdown commands. The following sections describe how you might use each of the shutdown commands:

Table 2-3 Shutdown Commands

Command	Purpose
`shutdown`	Use the `shutdown` command when shutting down a system with multiple users. The `shutdown` command sends a warning message to all users who are logged in, waits for 60 seconds (the default), and then shuts down the system to single-user state. You can choose a different default wait time (see "How to Change the Shutdown Grace Period" on page 25).
`init`	Use the `init` command to shut down a single-user system or to change its run level. You can use `init` to place the system in power-down state (`init 0`) or into single-user state (`init 1`).
`halt`	Use the `halt` command when the system must be stopped immediately and it is acceptable not to warn any current users. The `halt` command shuts down the system without any delay. It does not warn any other users on the system.
`reboot`	Use the `reboot` command to shut down a single-user system and bring it into multiuser state. `reboot` does not warn other users on the system.

Note – `init` and `shutdown` are the most reliable ways to shut down a system because they use `rc` scripts to kill running processes and shut down the system with minimal data loss. The `halt` and `reboot` commands do not run the `rc` scripts properly and are not the preferred method for shutting down the system. See Chapter 3, "Boot Files," for more information about the `rc` scripts.

Shutting Down a System by Using shutdown

This section provides examples of how to use the shutdown command to shut down a system.

▼ How to Shut Down a Multiuser System

Before shutting down a multiuser system, notify users and give them time to complete critical procedures.

1. **Type** who **and press Return.**
   A list of all logged-in users is displayed. You may want to send mail or broadcast a message to let users know that the system is being shut down.
2. **Become superuser.**
3. **Type** cd / **and press Return.**
   You must be in the root directory to run the shutdown command.
4. **Type** shutdown **and press Return.**
   A message is broadcast to all users. After a 60-second wait, you are asked to confirm that you want to shut down the system.
5. **Type** y **and press Return.**
   The system is shut down to single-user state and you are prompted for the root password.
6. **Type the root password.**
   The system is in single-user state and you can perform maintenance tasks.
7. **Press Control-d to return to the default run system level.**
   The following example shows the messages that may result when you use the shutdown command.

```
# cd /
# shutdown
Shutdown started Fri Jan 14 10:50:35 EDT 1994

Broadcast message from root (console) on earth Fri Aug 9 10:59:35.
THE SYSTEM IS BEING SHUT DOWN NOW ! ! !
LOG OFF NOW OR RISK YOUR FILES BEING DAMAGED
Do you want to continue?  (y or n):  y
```

```
The system is down.
Changing to init state s – please wait.

INIT: New run level S
INIT: SINGLE USER MODE
Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance):
```

▼ How to Shut Down and Reboot a Multiuser System

1. **Become superuser.**
2. **Type** `cd /` **and press Return.**
   You must be in the `root` directory to run the `shutdown` command.
3. **Type** `shutdown -i6` **and press Return.**
   A message is broadcast to all users. Then the `rc6` script is executed, the system is shut down to power-down state, and you can then bring it back up to multiuser state.

Overriding the `shutdown` *Defaults*

The following examples show how to change the default actions of the `shutdown` command.

▼ How to Shut Down a System Without Confirmation

1. **Become superuser.**
2. **Type** `cd /` **and press Return.**
   You must be in the `root` directory to run the `shutdown` command.
3. **Type** `shutdown -y` **and press Return.**
   The shutdown proceeds without asking you to type `y` to confirm it.

▼ How to Change the Shutdown Grace Period

1. **Become superuser.**
2. **Type** `cd /` **and press Return.**
   You must be in the `root` directory to run the `shutdown` command.
3. **Type** `shutdown -g`*nnn* **and press Return.**
   The grace period is changed to the number of seconds you specify.

This example changes the grace period to 120 seconds.

```
# cd /
# shutdown -g120
```

Shutting Down a System by Using `init`

▼ How to Shut Down a Single-User System

- **Type** `init 0` **and press Return.**
  The `init` command runs scripts that bring the system down cleanly. No warning messages are broadcast.

▼ How to Shut Down and Reboot a Single-User System

- **Type** `init 6` **and press Return.**
  This command writes data to the disk, kills all active processes, brings the system down to power-down state, and displays the PROM prompt. You can then reboot the system to the default level (usually multiuser).

▼ How to Shut Down Your System Quickly

To shut down a system quickly:

- **Type** `uadmin 2 0` **and press Return.**
  Data is written to the disk and the system is brought to power-down state. The system displays the PROM prompt.

2

Boot Files 3

This chapter contains the following sections:

Booting is the process of starting your system when it is first switched on or when it is restarted after a halt or shutdown. This chapter describes the files used at boot time. See *Administration Supplement for Solaris Platforms* for details about the boot procedure and instructions for booting.

Kernel Modules Directory (`/kernel`*)*

The SunOS kernel consists of a small static core and many dynamically loadable kernel modules. Many kernel modules are loaded automatically at boot time. Others, such as device drivers, are loaded in from disk as needed by the kernel.

The following directories are in `/kernel`:

```
pluto% ls -l /kernel
total 1760
drwxr-xr-x   2 root     sys          1536 Feb  5 16:32 drv
drwxr-xr-x   2 root     sys           512 Feb  5 16:27 exec
drwxr-xr-x   2 root     sys           512 Feb  5 16:27 fs
drwxr-xr-x   2 root     sys           512 Feb  5 16:29 misc
drwxr-xr-x   2 root     sys           512 Feb  5 16:27 sched
drwxr-xr-x   2 root     sys           512 Feb  5 16:31 strmod
drwxr-xr-x   2 root     sys           512 Feb  5 16:28 sys
-rwxr-xr-x   1 root     sys        881644 Jan 17 19:36 unix
pluto%
```

Drivers
Executable types
File system drivers
Miscellaneous modules
Scheduling classes
STREAMS drivers
System calls
Operating system kernel

Kernel Configuration File (`/etc/system`)

The boot program contains a list of default kernel modules to be loaded. You can use the `/etc/system` configuration file, which is read at boot time, to override the list of default modules.

Use the `/etc/system` file to specify:

- Which modules are automatically loaded
- Which modules are not automatically loaded
- Root and swap types and devices
- Overrides for default values of any kernel integer variables

See an example the default `/etc/system` file in Chapter 5, “Examining and Changing System Information.”

The System Initialization File (`/etc/inittab`)

When you boot your system or change run levels with the `init` command, the `init` daemon starts processes by using the information read from the entries in `/etc/inittab`. This file defines system initialization states (also called run levels). See “Init States” on page 18 for a discussion of init states.

Each entry in the `/etc/inittab` file has the following fields:

id:*runlevel*:*action*:*process*

Table 3-1 describes the fields in an `inittab` entry.

Table 3-1 Fields in the `inittab` File

Field	Description
id	A unique identifier
runlevel	The run level
action	How the process is to be run
process	The name of the command to execute

The following example shows an annotated default `inittab` file:

Code Example 3-1 The Default `inittab` File

```
                ap::sysinit:/sbin/autopush -f /etc/iu.ap
File system check
                fs::sysinit:/sbin/bcheckrc              >/dev/console 2>&1 \
                </dev/console
Default run level
                is:3:initdefault:
Power-fail shutdown
                p3:s1234:powerfail:/sbin/shutdown -y -i0 -g0 >/dev/console 2>&1 \
                </dev/console
Init 0 run level
                s0:0:wait:/sbin/rc0 off                 >/dev/console 2>&1 \
                </dev/console
Init 1 run level
                s1:1:wait:/sbin/shutdown -y -iS -g0     >/dev/console 2>&1 \
                </dev/console
Init 2 run level
                s2:23:wait:/sbin/rc2                    >/dev/console 2>&1 \
                </dev/console
Init 3 run level
                s3:3:wait:/sbin/rc3                     >/dev/console 2>&1 \
                </dev/console
Init 5 run level
                s5:5:wait:/sbin/rc5 ask                 >/dev/console 2>&1 \
                </dev/console
Init 6 run level
                s6:6:wait:/sbin/rc6 reboot              >/dev/console 2>&1 \
                </dev/console
Off
                of:0:wait:/sbin/uadmin 2 0              >/dev/console 2>&1 \
                </dev/console
Firmware
                fw:5:wait:/sbin/uadmin 2 2              >/dev/console 2>&1 \
                </dev/console
Reboot
                RB:6:wait:/sbin/sh -c 'echo "\nThe system is being restarted."'\
                >/dev/console 2>&1
```

Code Example 3-1 The Default `inittab` File (Continued)

```
Reboot single-user      rb:6:wait:/sbin/uadmin 2 1              >/dev/console 2>&1 \
                        </dev/console
Service access controller sc:234:respawn:/usr/lib/saf/sac -t 300
Console                 co:234:respawn:/usr/lib/saf/ttymon -g -h -p "‘uname -n‘ \
                        console  login: " -T sun -d /dev/console -l console \
                        -m ldterm,ttcompat
```

When the system is first booted, `init` starts all processes labeled `sysinit` in the `inittab` file. The `initdefault` entry in `/etc/inittab` identifies the default run level. In this example, the default is run level 3 (multiuser mode with network file sharing). The `init` daemon runs each process associated with this run level (each entry that has a `3` in its `rstate` field). Each process is run using the entry from the action field. The action field can have one of the following values:

- `powerfail` – The system has received a powerfail signal
- `wait` – Wait for the command to be completed
- `respawn` – Restart the command

In this example, the commands executed at run level 3 are:

- `/usr/sbin/shutdown`
  This command shuts down the system. `init` runs the `shutdown` command only if the system has received a `powerfail` signal.
- `/sbin/rc2`
  This command sets up the time zone, then starts the standard system processes, bringing the system up into run level 2 (multiuser mode).
- `/sbin/rc3`
  This command starts NFS.
- `/usr/lib/saf/sac -t 30`
  This command starts the port monitors and net access for UUCP.
- `/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: " -T` *terminal_type* `-d /dev/console -l console`
  `ttymon` monitors the console for login requests.

The `/usr/lib/saf/sac` and `/usr/lib/saf/ttymon` processes are respawned (restarted) if they are terminated for any reason.

Run Control Files

The SunOS operating system provides a detailed series of run control (`rc`) scripts to control init state changes. Each run level (also called an init state) has an associated `rc` script located in the `/sbin` directory.

The following run control scripts are in the `/sbin` directory:

- `rc0`
- `rc1`
- `rc2`
- `rc3`
- `rc5`
- `rc6`
- `rcS`

For each `rc` script in the `/sbin` directory, there is a corresponding directory named `/etc/rc`*n*`.d` that contains scripts to perform various actions for that run level. For example, `/etc/rc2.d` contains files used to start and stop processes for run level 2.

```
saturn% ls /etc/rc2.d
K20lp           S21perf         S72inetsvc      S89bdconfig
K60nfs.server   S30sysid.net    S73nfs.client   S90xtl
K65nfs.client   S47asppp        S74autofs       S91gsconfig
K92volmgt       S69inet         S74syslog       S91gtconfig
README          S70uucp         S75cron         S91leoconfig
S01MOUNTFSYS    S71rpc          S80PRESERVE     S92rtvc-config
S05RMTMPFILES   S71sysid.sys    S80lp           S92volmgt
S18setuname     S72autoinstall  S88sendmail     S99audit
S20sysetup
saturn%
```

Run control scripts are also located in the `/etc/init.d` directory. These files are linked to corresponding run control scripts in the `/etc/rc*.d` directories.

The `/etc/rc`*n*`.d` scripts are always run in ASCII sort order. The scripts have names of the form:

`[K,S][0-9][0-9][A-Z][0-99]`

Files beginning with `K` are run to terminate (kill) some system process. Files beginning with `S` are run to start up a system process.

The actions of each run control level script are summarized as follows.

The /sbin/rc0 *Script*

- Stops system services and daemons
- Terminates all running processes
- Unmounts all file systems

The /sbin/rc1 *Script*

- Runs the /etc/rc1.d scripts
  - Stops system services and daemons
  - Terminates all running processes
  - Unmounts all file systems
  - Brings the system up in single-user mode

```
pluto% ls /etc/rc1.d
K00ANNOUNCE    K55syslog      K67rpc         K80nfs.client
K42audit       K57sendmail    K68autofs      S01MOUNTFSYS
K47asppp       K65nfs.server  K70cron
pluto%
```

The /sbin/rc2 *Script*

See the *Administration Supplement for Solaris Platforms* for information about scripts that are platform-specific.

- Sets the TIMEZONE variable
- Runs the /etc/rc2.d scripts
  - Mounts all file systems
  - Enables disk quotas if at least one file system was mounted with the quota option
  - Saves editor temporary files in /usr/preserve
  - Removes any files in the /tmp directory
  - Creates device entries in /dev for new disks (only if boot -r is run)
  - Prints system configuration (the default is not to save a core file)
  - Configures system accounting
  - Configures default router
  - Sets NIS domain
  - Sets ifconfig netmask
  - Starts inetd

- Starts `named`, if appropriate
- Starts `rpcbind`
- Starts Kerberos client-side daemon, `kerbd`
- Starts NIS daemons (`ypbind`) and NIS+ daemons (`rpc.nisd`), depending on whether the system is configured for NIS or NIS+, and whether the system is a client or a server
- Starts `keyserv`
- Starts `statd`, `lockd`
- Mounts all NFS entries
- Starts `automount`
- Starts `cron`
- Starts the LP daemons
- Starts the `sendmail` daemon

```
saturn% ls /etc/rc2.d
K20lp           S21perf         S72inetsvc      S89bdconfig
K60nfs.server   S30sysid.net    S73nfs.client   S90xtl
K65nfs.client   S47asppp        S74autofs       S91gsconfig
K92volmgt       S69inet         S74syslog       S91gtconfig
README          S70uucp         S75cron         S91leoconfig
S01MOUNTFSYS    S71rpc          S80PRESERVE     S92rtvc-config
S05RMTMPFILES   S71sysid.sys    S80lp           S92volmgt
S18setuname     S72autoinstall  S88sendmail     S99audit
S20sysetup
saturn%
```

The /sbin/rc3 *Script*

- Runs the /etc/rc3.d scripts
  - Cleans up sharetab
  - Starts nfsds
  - Starts mountd
  - If boot server, starts rarpd and rpc.bootparamd

```
saturn% ls /etc/rc3.d
README          S15nfs.server
saturn%
```

The /sbin/rc5 *Script*

- Runs the /etc/rc0.d scripts
  - Kills the printer daemons
  - Unmounts local file systems
  - Kills the syslog daemon
  - Unmounts remote file systems
  - Stops NFS services
  - Stops NIS services
  - Stops RPC services
  - Stops cron services
  - Stops NFS client services
- Kills all active processes
- Initiates an interactive boot

```
saturn% ls /etc/rc0.d
K00ANNOUNCE     K47asppp        K66nfs.server   K75nfs.client
K20lp           K55syslog       K69autofs       K85rpc
K42audit        K57sendmail     K70cron
saturn%
```

The /sbin/rc6 *Script*

- Runs /etc/rc0.d/K*
- Kills all active processes
- Unmounts the file systems
- Runs the initdefault entries in /etc/inittab

The /sbin/rcS *Script*

- Runs the /etc/rcS.d scripts to bring the system up to single user mode
  - Establishes a minimal network
  - Mounts /usr, if necessary
  - Sets the system name
  - Checks the / and /usr file systems
  - Checks and mounts the /usr/kvm file system, if necessary
  - Mounts pseudo file systems (/proc and /dev/fd)
  - For reconfiguration boots, rebuilds the device entries
  - Checks and mounts other file systems to be mounted in single user mode

```
saturn% ls /etc/rcS.d
README                S40standardmounts.sh  S60devlinks
S30rootusr.sh         S50drvconfig          S70buildmnttab.sh
saturn%
```

Using the Run Control Scripts to Stop or Start Services

One advantage of individual scripts for each run control state is that you can run scripts in the /etc/init.d directory individually to turn off an area of functionality for a system without changing its run control level. For example, you can turn off NFS server functionality by typing /etc/init.d/nfs.server stop and pressing Return. After you have changed the system configuration, you can restart the functionality by typing /etc/init.d/nfs.server start and pressing Return.

Adding Scripts to the Run Control Directories

If you add a script, put the script in the /etc/init.d directory and create a link to the appropriate rc*.d directory. Assign appropriate numbers and names to the new scripts so that they will be run in the proper sequence.

If you rename a file and do *not* want the renamed file to be run, use a dot (.) at the beginning of the new file name. Files that begin with a dot are not executed. If you copy a file, adding a suffix to it, both files will be run.

For example, if you want to change the K00ANNOUNCE script and save the original script, type:

```
# cp K00ANNOUNCE .K00ANNOUNCE
#
```

If you copy K00ANNOUNCE to K00ANNOUNCE.old, for example, both K00ANNOUNCE and K00ANNOUNCE.old are run when the system is shutdown.

Understanding the Boot Messages File

During the start-up process, messages are displayed. These messages report the results of the boot procedure, including information about what hardware is found in the system, the amount of memory available, and any problems encountered during the boot process. The information in these messages is stored in the /var/adm/messages file.

▼ How to Look at the Boot Messages

The most recent boot messages are stored in the /var/adm/messages file. To see these messages after you have booted the system:

- **Type** /usr/sbin/dmesg **and press Return.**
  The boot messages are displayed.

  or

- **Type** more /var/adm/messages **and press Return.**

See the *Administration Supplement for Solaris Platforms* for an example of the /var/adm/messages file.

Adding Systems to a Network 4

This chapter contains the following sections:

System Configurations

There are five system configurations. These are:

- Servers
- Standalone networked systems
- Standalone non-networked systems
- Diskless clients
- Dataless clients

4

Servers

A *server* reserves a portion of its disk space for use by other machines, called *clients.* A server typically has a large amount of disk storage capacity, and so can afford to share some of its disk space. A client usually has limited disk storage capacity, or perhaps none at all.

A server system has the following file systems:

- The `root` and `/usr` file systems, plus swap space
- The `/export`, `/export/swap`, and `/export/home` file systems, which support client systems and provide home directories for users.
- The `/opt` directory or file system for storing application software.

Note – Strictly speaking, a server is any machine that provides a service to other machines in its network. There are file servers, boot servers, database servers, license servers, print servers, media servers, and even servers for particular applications. However, except where otherwise noted, this chapter uses the term server to mean a machine that provides disk space for the storage of other machines' files and programs.

Standalone Networked Systems

A *standalone networked system* can share information with other machines in the network, but it could continue to function if detached from the network.

A standalone system can function autonomously because it has its own hard disk containing `root`, `swap`, `/usr`, and `/home` file systems. The standalone system thus has local access to operating system software, executables, virtual memory space, and user-created files.

Note that a standalone system requires sufficient disk space—usually 150 Mbytes or more—to hold the four necessary file systems.

Some system administrators implement a *pseudo-standalone* configuration using a server to store user-created files. Note however, that if denied access to the user file system across the net, the pseudo-standalone machine has very limited capabilities.

Standalone Non-Networked Systems

A standalone non-networked system is essentially the same as a standalone system that is not connected to the network. A *standalone non-networked system* is used strictly on its own (that is, not connected to any network).

Diskless Clients

A *diskless client* has no disk and depends on a server for all its software and storage area. A diskless client remotely mounts its `/`, `/usr`, and `/home` file systems from a server.

A diskless client generates significant network traffic due to its continual need to procure operating system software and virtual memory space from across the network. A diskless client cannot operate if it is detached from the network or if its server malfunctions.

Dataless Clients

A *dataless client* has local storage for its `root` file system and swap space. The dataless client cannot function if detached from the network, because its executables (`/usr`) and user files (`/home`) are located across the network on the disk of a server.

A dataless client places far less demand on the server and the network than a diskless client does.

Because dataless clients require less network access, a server can accommodate many more dataless clients than it can diskless clients.

Dataless clients are cheaper than standalone systems. Also, the user files of all the dataless clients are stored centrally (on a server) and can be backed up and administered centrally.

However, if local security is an issue at your site, you need to weigh these conveniences against the relative lack of privacy of centralized files.

4

Using Host Manager to Add Clients and Standalone Systems

Host Manger is an Administration Tool application which uses a graphical interface to manage7 network client information. See *Administration Application Reference Manual* for more information on Host Manager.

Before using Host Manager, verify that the following requirements are met:

- A bit-mapped display monitor is connected to the system you are using.
- The OpenWindows™ environment is running. If needed, use the following command to start the OpenWindows environment:

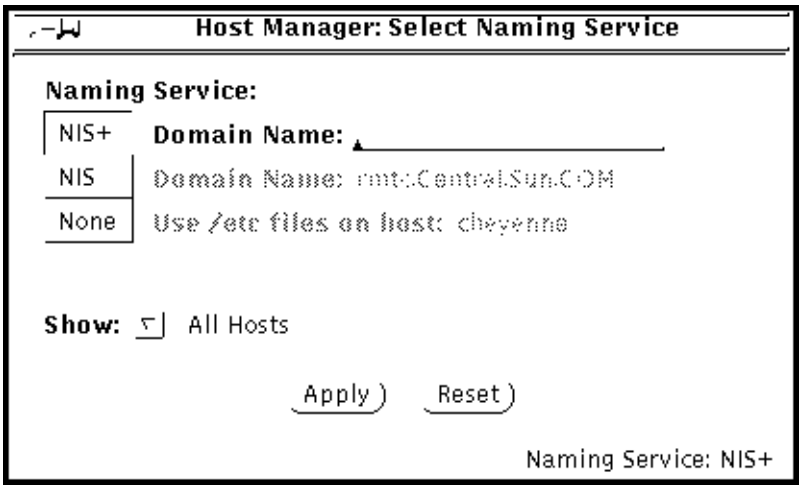
```
$ /usr/openwin/bin/openwin
```

- You have the required access privileges such as root (superuser) access to the local system or membership in the sysadmin (group ID=14) group for remote systems. If you are using Host Manager to update a NIS or NIS+ file, you must be a member of the sysadmin group and have the appropriate access privileges.

▼ How to Start Host Manager

The following procedure describes how to start Host Manager.

1. **Type** `admintool &` **from a Command or Shell Tool prompt and press Return.**
   The Administration Tool main window is displayed.
2. **Click on the Host Manager icon.**
   The Select Naming Service window is displayed.
3. **Select the name service used in your network.**
4. **Check that the domain or host name is correct.**
   If not, type the domain or host name you need to access.
5. **Click on Apply.**
   The Host Manager main window is displayed.


Host Manager: Select Naming Service
Naming Service:
NIS+
Domain Name:
NIS
Domain Name: mnt.CentralSun.COM
None
Use /etc files on host: cheyenne
Show:
All Hosts
Apply
Reset
Naming Service: NIS+

▼ How to Add Standalone System Information

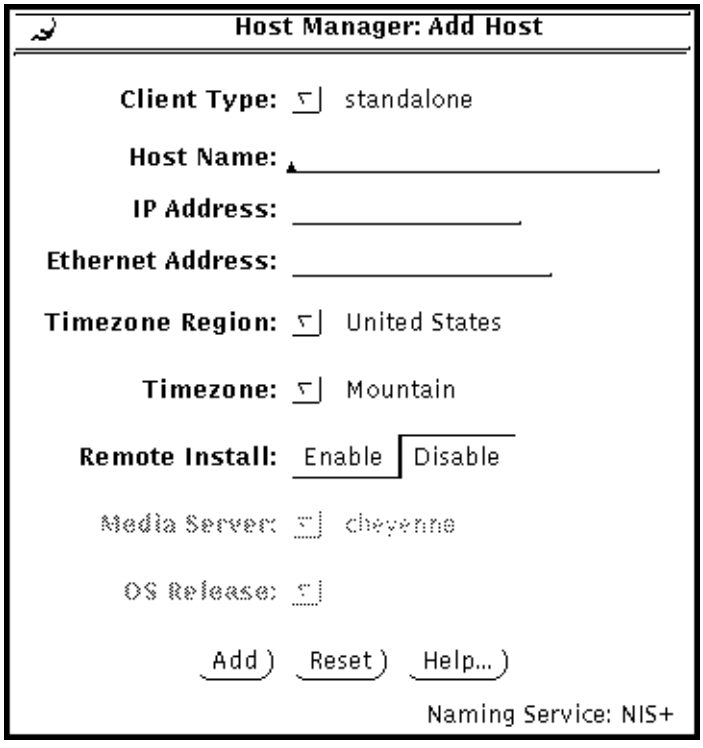
Use these steps to add standalone system information. The following items must be provided for the standalone system:

- Host name
- IP address
- Ethernet address

Default settings are available for:

- Time zone region and time zone
- Media server

1. **Select Add Host from the Edit menu on the Host Manager Window.**
   The Add Host window is displayed.



2. **Fill in the Add Host window.**

3. **If this system will be installed remotely, follow Steps a and b to enable remote install privileges. Otherwise, skip to step 4.**

   a. **Click on Enable.**
   The Media Server defaults to the current host.

   b. **Select Other from the Media Server menu to specify another host as the media server, if necessary. Type the host name in the text box provided and click on Apply on the Media Server menu.**

4. **Click on Apply on the Add Host window.**

5. **Verify that the new host has been added by locating the new host entry in the Host Manager main window.**

▼ How to Add Support for a Diskless Client

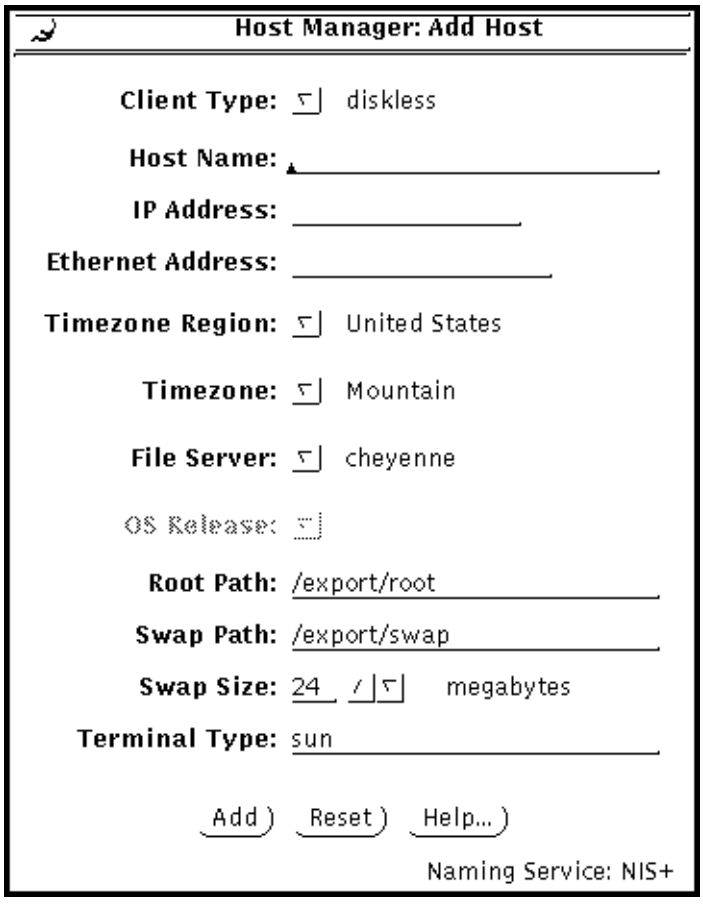
Use these steps to add support for a diskless client. This procedure assumes the system providing the services has already been configured as a server, meaning the `/export` and `/export/swap` file systems are already created. The following items must be provided for the diskless client:

- Host name
- IP address
- Ethernet address

Default settings are available for:

- Time zone region and time zone
- File server
- OS release
- Root and swap paths
- Swap size
- Terminal type

1. **Select Add Host from the Edit menu on the Host Manager main window.**
   The Add window is displayed.



2. **Select diskless from the Client Type menu.**
3. **Fill in the Add Host window.**
4. **If another host will act as the file server for this system, follow Steps a and b. Otherwise, skip to step 5.**
   a. **Select Other from the File Server menu to specify another host as the file server.**

b. Fill out the host information in the text box provided and click on Apply on the Specify File Server window.

5. **Click on Add on the Add Host window.**
   It takes several minutes to add the diskless client support, particularly to create the client's root and swap areas.

6. **You may need to reboot the file server to start the appropriate client daemons after the client information has been added successfully.**
   A message about rebooting the system will be displayed.

7. **Boot the diskless client system.**

```
# boot net
```

8. **Provide the following system configuration information for the diskless client during the initial boot process, if prompted.**
   - Geographic region
   - Time zone
   - Date and time

9. **Create a root password when prompted.**

▼ How to Add Support for a Dataless Client

Use these steps to add support for a dataless client on a server. This procedure assumes the following tasks have already been completed:

- The system providing the services has already been configured as a server, meaning the `/export` file system is already created.
- The dataless client system has already been configured using the Solaris installation program. See the Solaris software installation manual for your platform for information about using this program.

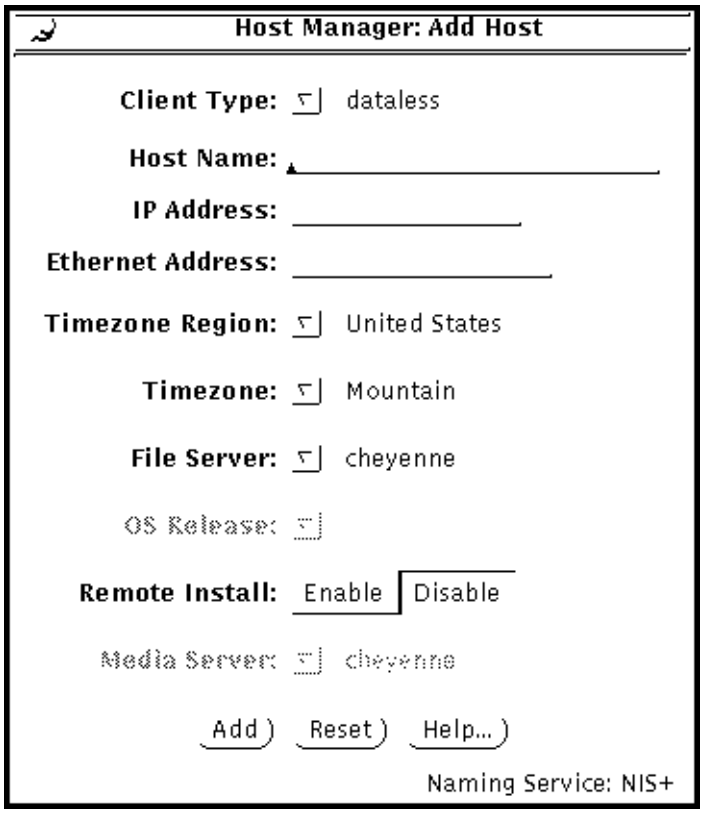
  The following information must be provided for the dataless client:

  - Host name
  - IP address
  - Ethernet address

  Default settings are also available for:

  - Time zone region and time zone
  - File server
  - Operating system release
  - Remote install
  - Media server

1. **Select Add Host from the Edit menu on the Host Manager main window.**
   The Add window is displayed.



2. **Select dataless from the Client Type menu.**

3. **Fill in the Add Host window.**

4. **Click on Add.**
   It takes several minutes for the dataless client support to be added.

4

Examining and Changing System Information 5

This chapter contains the following sections:

Examining System Information

This section describes how to examine system information such as host name, amount of memory, or IP address.

Determining the PROM Version

Refer to *Administration Supplement for Solaris Platforms* for instructions on determining the PROM version of a system.

▼ How to Determine the Host Name

♦ **Type** `uname -n` **and press Return.**

```
# uname -n
jupiter
#
```

▼ How to Print the Host ID number

♦ **Type** `sysdef -h` **and press Return.**

```
# sysdef -h
*
* Hostid
*
 51025dec
#
```

The host ID is shown in hexadecimal format.

▼ How to Determine the Amount of Memory

♦ **Type** `prtconf | grep Memory` **and press Return.**

```
# prtconf | grep Memory
Memory size: 28 Megabytes
#
```

▼ How to Determine the Kernel Architecture Type

♦ **Type** `uname -m` **and press Return.**

```
# uname -m
sun4c
#
```

▼ How to Determine the Processor Type

♦ **Type** `uname -p` **and press Return.**

```
# uname -p
sparc
#
```

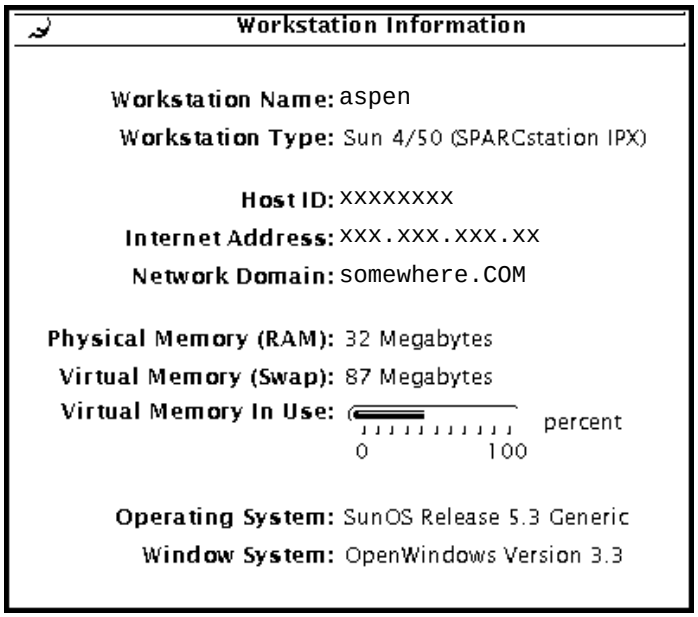
▼ How to Determine the Operating System Release

♦ **Type** `uname -r` **and press Return.**

```
# uname -r
5.4
#
```

▼ How to Display System Information With the Workspace Menu

♦ **Choose Workstation Info from the OpenWindows Workspace menu.**
A window similar to the following is displayed.



5

The /etc/system *File*

The /etc/system file is used to customize the Solaris operating system kernel. The commands in the system file are read by the kernel during initialization at boot time.

Format of the /etc/system *File*

The /etc/system file is a list of commands that consist of keyword and value pairs. A command line must be 80 characters or less in length. Comment lines must begin with an asterisk and end with a newline character. The following command sets the maximum number of users that can be logged in to the system to 40:

```
set maxusers=40
```

Commands can modify the loadable kernel modules, root device, root file system type, swap device, swap file system type, and kernel parameters. For information on the commands that can be used in the /etc/system file, see the system(4) manual page.

For more information on modifying kernel parameters, see "Changing System Information" on page 54. For a list of configurable kernel parameters, see *Security, Performance, and Accounting Administration*.

The following example shows the default /etc/system file.

Table 5-1 The Default /etc/system File

```
*ident"@(#)system1.1592/11/14 SMI" /* SVR4 1.5 */
*
* SYSTEM SPECIFICATION FILE
*

* moddir:
*
*   Set the search path for modules.  This has a format similar to the
*   csh path variable. If the module isn't found in the first directory
*   it tries the second and so on. The default is /kernel /usr/kernel
*
```

Table 5-1 The Default `/etc/system` File (Continued)

```
*   Example:
*       moddir: /kernel /usr/kernel /other/modules

* root device and root filesystem configuration:
*
*   The following may be used to override the defaults provided by
*   the boot program:
*
*   rootfs: Set the filesystem type of the root.
*


*
*   rootdev:Set the root device.  This should be a fully
*           expanded physical pathname.  The default is the
*           physical pathname of the device where the boot
*           program resides.  The physical pathname is
*           highly platform and configuration dependent.
*
*   Example:
*       rootfs:ufs
*       rootdev:/sbus@1,f8000000/esp@0,800000/sd@3,0:a
*
*   (Swap device configuration should be specified in /etc/vfstab.)

* exclude:
*
*   Modules appearing in the moddir path which are NOT to be loaded,
*   even if referenced. Note that ‘exclude' accepts either a module name,
*   or a filename which includes the directory.

*
*   Examples:
*       exclude: win
*       exclude: sys/shmsys

* forceload:
*
```

Table 5-1 The Default `/etc/system` File (Continued)

```
*   Cause these modules to be loaded at boot time, (just before mounting
*   the root filesystem) rather than at first reference. Note that
*   forceload expects a filename which includes the directory. Also
*   note that loading a module does not necessarily imply that it will
*   be installed.
*
*   Example:
*       forceload: drv/foo

* set:
*
*   Set an integer variable in the kernel or a module to a new value.
*   This facility should be used with caution.  See system(4).
*
*   Examples:
*
*   To set variables in 'unix':
*
*       set nautopush=32
*       set maxusers=40
*
*   To set a variable named 'debug' in the module named 'test_module'
*
*       set test_module:debug = 0x13
```

Changing System Information

Some system changes require modifying the `/etc/system` file. Before editing the `/etc/system` file, you should save a copy of the original file.

▼ How to Set the Number of Processes per User

1. **Edit the `/etc/system` file with a text editor and add the following line:**
   `set maxuprc=`***value***
   where *value* is the number of processes a user can run at one time.

2. **Type** `touch /reconfigure` **and press Return.**
3. **Reboot the system.**

▼ How to Change the Host Name of a System

1. **Edit the following files with a text editor, changing the host name in each file:**
   - `/etc/nodename`
   - `/etc/hostname.`*ethernet_device_interface* (for example, `/etc/hostname.le0` or `/etc/hostname.smc0`)
   - `/etc/inet/hosts`
   - `/etc/net/ticlts/hosts`
   - `/etc/net/ticots/hosts`
   - `/etc/net/ticotsord/hosts`
2. **Reboot the system.**

▼ How to Increase the Number of Pseudo-ttys to 256

1. **Edit the** `/etc/system` **file with a text editor and add the following line:**

   ```
   set pt_cnt=256
   ```

2. **Type** `touch /reconfigure` **and press Return.**
3. **Reboot the system.**

▼ How to Increase the Number of Lock Requests

The default number of lock requests that may occur simultaneously is 512. As users log out, they lock files, including `utmp`. If more than 512 users are likely to log out simultaneously (within a few seconds), the number of file locks allowed must be increased, as follows:

1. **Edit the** `/etc/system` **file with a text editor and add the following line:**

   ```
   set tune_t_flckrec=1024
   ```

2. **Type** `touch /reconfigure` **and press Return.**
3. **Reboot the system.**

▼ How to Increase Shared Memory Segments

The following instructions show how to increase the kernel parameter values for shared memory. The values used are appropriate for a system with a large amount of memory (for example, 128 MBytes) that is running a large database application.

1. **Edit the** `/etc/system` **file with a text editor and add the following lines:**

```
set shmsys:shminfo_shmmax=268435456
set semsys:seminfo_semmap=250
set semsys:seminfo_semmni=500
set semsys:seminfo_semmns=500
set semsys:seminfo_semmsl=500
set semsys:seminfo_semmnu=500
set semsys:seminfo_semume=100
set shmsys:shminfo_shmmin=200
set shmsys:shminfo_shmmni=200
set shmsys:shminfo_shmseg=200
```

2. **Type** `touch /reconfigure` **and press Return.**

3. **Reboot the system.**

Using crontab 6

This chapter describes how to use the crontab command to submit commands to be run at a later time; how to edit, display, and remove crontab files; and how to restrict access to the crontab facility. This chapter contains the following sections:

Overview

There are two parts to the crontab facility.

- The crontab command allows you to submit commands to be run later. You can also have the command run at regular intervals.
- The cron command starts the cron daemon, which runs the commands submitted with crontab.

You can use the crontab facility to automate tasks such as backups or other time consuming tasks.

6

The `crontab` *Command*

To submit a command to be executed later, you use the `crontab` command to create a `crontab` file. The `crontab` file specifies what the command is, when it should be run, and how often. See "The Format of crontab Files" on page 58 for a description of `crontab` files. See "Creating crontab Files" on page 59 for instructions to create a `crontab` file.

The `at(1)` command can be used to submit a command that will only be executed once. See the `at(1)` manual page for more information.

The `cron` *Command*

The `cron` command is normally executed by an `/sbin/rc2.d` script at boot time. The `cron` daemon executes the commands in the `crontab` file at the specified times. See the `cron(1M)` manual page for more information on the `cron` daemon.

The Format of `crontab` *Files*

A line in a `crontab` file consists of the following six fields, which are separated by spaces:

minute hour day_of_the_month month day_of_the_week command

The fields have the following meanings:

- *minute*: The minute at which to execute the specified command, ranging from `0`, for on the hour, to `59`, for one minute before the hour
- *hour*: The hour, based on a 24-hour clock, at which to execute the specified command, ranging from `0`, for midnight, to `23`, for 11 p.m.
- *day_of_the_month*: The day of the month on which to execute the specified command, ranging from `1` to `31`
- *month*: The month of the year during which to execute the command, ranging from `1` to `12`
- *day_of_the_week*: The day of the week to execute the specified command, ranging from `0` (for Sunday) to `6` (for Saturday)
- *command*: The command that should be run at the specified time

An asterisk indicates a field is left blank.

Note that the day can be specified by two of the fields (*day_of_the_month* and *day_of_the_week*). If both are specified, both will be used. If one of the two fields is an asterisk, only the other is used.

If the first five fields of a `crontab` line were:

```
0 0 15,30 * 1
```

the command would be run at midnight on the fifteenth and thirtieth of each month and on every Monday. To specify only specific days of the week to run a command, put an asterisk (*) in the *day_of_the_month* field. To specify only days of the month, put an asterisk (*) in the *day_of_the_week* field.

If you do not direct the standard output and standard error for the command, any output for standard output or standard error is mailed to you.

Any line in a `crontab` file that begins with a hash mark (#) is a comment and is ignored.

Creating `crontab` *Files*

▼ How to Create a `crontab` File

1. **Type** `crontab` **and press Return.**
   The `crontab` command expects the `crontab` file information from standard input.
2. **Type the** `crontab` **line and press Return.**
3. **Press Control-d to indicate the end of standard input.**

You can also create a `crontab` file by using a text editor to create a file that contains the `crontab` information and then specifying the name of the file as an argument to the `crontab` command.

To create a `crontab` file for the superuser, you must first become superuser and then create a `crontab` file.

6

Editing `crontab` *Files*

Use the `-e` option to `crontab` to edit `crontab` files. The environment variable `EDITOR` determines which editor is used. If the `EDITOR` environment variable has not been set, the default editor `ed(1)` is used.

You must be the superuser to edit the superuser's `crontab` file, or to edit a `crontab` file that belongs to another user.

▼ How to Edit Your Own `crontab` File

♦ **Type** `crontab -e` **and press Return.**

▼ How to Edit the Superuser's `crontab` File

1. **Become superuser.**
2. **Type** `crontab -e` **and press Return.**

▼ How to Edit Another User's `crontab` File

1. **Become superuser.**
2. **Type** `crontab -e` *username* **and press Return.**

Displaying `crontab` *Files*

You must be superuser to display the contents of the superuser's `crontab` file or to display a `crontab` file that belongs to another user.

Use the `-l` option to the `crontab` command to display a `crontab` file.

Caution – If you inadvertently enter the `crontab` command with no argument, do *not* attempt to exit by pressing Control-d which would remove all entries in the `crontab` file. Instead, exit by pressing the interrupt character (usually Control-c).

▼ How to Display Your Own `crontab` File

♦ **Type** `crontab -l` **and press Return.**

▼ How to Display the Superuser's `crontab` File

1. **Become superuser.**
2. **Type** `crontab -l` **and press Return.**

▼ How to Display a User's `crontab` File

1. **Become superuser.**
2. **Type** `crontab -l` *username* **and press Return.**

Removing `crontab` *Files*

You must be superuser to remove a `crontab` file that belongs to another user or to remove the superuser's `crontab` file.

Use the `-r` option to the `crontab` command to remove a `crontab` file.

▼ How to Remove Your Own `crontab` File

♦ **Type** `crontab -r` **and press Return.**

▼ How to Remove the Superuser's `crontab` File

1. **Become superuser.**
2. **Type** `crontab -r` **and press Return.**

▼ How to Remove a User's `crontab` File

1. **Become superuser.**
2. **Type** `crontab -r` *username* **and press Return.**

6

Controlling Access to `crontab`

The `cron.allow` and `cron.deny` files are used to control access to the `crontab` file. Both `cron.allow` and `cron.deny` are text files that you create and modify with a text editor. Both files are found in the directory `/etc/cron.d`.

The `cron.allow` file contains the names of users that are allowed to submit `crontab` files. The `cron.deny` file contains the names of users that cannot submit `crontab` files. Table 6-1 describes how to allow access to the `crontab` command in different cases.

Table 6-1 Creating `cron.deny` and `cron.allow`

To Allow These People to Submit `crontab` Files...	**Do the Following**
Superuser only	Delete (or do not create) `cron.deny` and `cron.allow`
All users	Create an empty `cron.deny` file. Do not create `cron.allow`.
No users, with a few exceptions	Create a `cron.allow` file containing a list of users who are allowed to submit `crontab` files. Do not create `cron.deny`.
All users, with a few exceptions	Create a `cron.deny` file containing a list of all users who cannot submit `crontab` files. Do not create `cron.allow`.

The `cron.deny` file is installed as part of the SunOS installation and the `cron.allow` file should be created by the superuser to extend access to the `crontab` command.

The following example shows the default `cron.deny` file:

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

Accessing Remote Files and Systems 7

This chapter has the following sections:

This chapter describes the commands you can use to access other systems on the network and to copy files or directories to or from remote systems.

Logging In to a Remote System

You use the `rlogin(1)` command to log in to other systems on the network.

▼ How to Log In to a Remote System

1. **Type** `rlogin` *system_name* **and press Return.**
   The system displays a password prompt if the remote system requires a password to log in.
2. **Type the password for your account on the remote system (if required) and press Return.**
   The command prompt for the remote system is displayed.

▼ How to Log Out From a Remote System

♦ **Type** `exit` **and press Return.**

▼ How to Log In to a Remote System With a Different User Name

1. **Type** `rlogin -l` *username system_name* **and press Return.**
   The password prompt is displayed if the remote system requires a password for the specified username.
2. **Type the password for the account on the remote system (if required) and press Return.**
   The command prompt for the remote system is displayed.

▼ How to Allow Remote Login by the Superuser

1. **Type** `rlogin` *system_name* **and press Return.**
2. **Type** `su` **and press Return.**
3. **Type the root password and press Return.**
4. **Edit the file** `/etc/default/login`**.**
   Put a hash mark (#) at the beginning of the following line:

```
CONSOLE=/dev/console
```

   This makes the line a comment and `CONSOLE` will not be set. If `CONSOLE` is set, the superuser can log in only on the console. You can now log in to the system as superuser from a remote system.

5. **Type** `exit` **and press Return to exit the superuser shell.**
6. **Type** `exit` **and press Return to log out from the remote system.**

Executing Commands on a Remote System Without Logging In

Use the `rsh(1)` (remote shell) command to execute a command on a remote system. `rsh` cannot be used to run interactive commands on a remote system; you must log in to the system to run interactive commands.

▼ How to Execute a Command on a Remote System

♦ **Type** `rsh` *system_name command* **and press Return.**

If the user from the local system does not have permission to execute remote commands on the remote system, the local system displays one of the following messages:

```
Permission denied
Login incorrect
```

The following example executes the `uptime` command on the system mars:

```
% rsh mars uptime
  4:58pm  up 5 day(s),  7:25,3 users,  load average: 0.04, 0.01, 0.02
```

Checking Who Is Logged In to a Remote System

You can check who is logged in to a remote system with the `rusers` command.

The `rusers` command lists remote systems followed by a list of all the users logged in on the system. The `rusers -l` option lists the following for all users logged in to remote systems:

- user name
- shell window
- login time and date
- amount of time logged in
- remote machine from which the user logged in

▼ How to Find Out Who Is Logged In to Remote Systems

1. **Type** `rusers -l` [*system_name*] **and press Return.**
   If you do not specify a system name, users logged on to all the systems on the network are shown.

The output of the `rusers -l` command is similar to the following:

```
jane      pluto:console       Nov 18 09:19
darcy     mars:ttyp1          Nov 17 10:17        8 (jupiter)
```

Copying a File or a Directory Over a Network

You can copy a file or directory from one system on the network to another with one of the following commands:

- `rcp` (remote copy)—Copies files or directories in either direction between your local system and a remote system on your network.
- `ftp` (file transfer)—Copies files (but not directories) in either direction between local and remote systems, when the systems are on different networks.

Note – To copy a file, you must have read permission. To copy a directory, you must have read and execute permission.

Copying a File or Directory With the `rcp` *Command*

Use the `rcp` command to copy a file or directory between local and remote systems on the same network.

Caution – Copying a file between a local and a remote system overwrites a file with the same name on the destination system. However, copying directories between local and remote systems does not overwrite identically named directories; instead the copied directory becomes a subdirectory within the identically named directory.

▼ How to Copy a Directory to a Remote System

♦ **Type** `rcp -r` *directory_name remote_system_name*`:[`*remote_directory_name*`]` **and press Return.**
If you do not specify a name for the directory on the remote system, it will have the same name as the directory on the local system.

The following example shows how to copy a directory to a remote system:

```
% rcp -r doc jupiter:docdir
```

▼ How to Copy a File to a Remote System

♦ **Type** `rcp` *filename remote_system_name*`:[`*remote_filename*`]` **and press Return.**
If you do not specify a name for the file on the remote system, it will have the same name as the file on the local system.

The following example copies a file to a remote system:

```
% rcp main.c mars:
```

▼ How to Copy a Directory From a Remote System

♦ **Type** `rcp -r` *remote_system_name*`:`*directory_name* `[`*local_directory_name*`]` **and press Return.**
If you do not specify *local_directory_name*, the directory that is copied to the local system will have the same name as the directory on the remote system.

This example copies a directory from a remote system to the current directory:

```
% rcp -r saturn:docdir .
```

▼ How to Copy a File From a Remote System

♦ **Type** `rcp` *remote_system_name*`:`*filename* `[`*local_filename*`]`
If you do not specify *local_filename*, the file that is copied to the local system will have the same name as the file on the remote system.

The following example shows how to copy a file from a remote system to the current directory:

```
% rcp mars:test.c .
```

Copying a File With the `ftp` *Command*

To copy files from your local system to a remote system, or from a remote system to your local system, you can use the `ftp` command. You cannot copy directories with `ftp`.

The `ftp` command is especially useful for transferring files between local and remote systems on different networks. See the `ftp(1)` manual page for complete details about the `ftp` command.

Caution – Copying a file between systems overwrites a file with the same name on the destination system.

▼ How to Copy a File to a Remote System

1. **Type** `ftp` *remote_system_name* **and press Return.**
2. **Type the user name for your account on the remote system and press Return.**
3. **Type the password for your account on the remote system (if required) and press Return.**
4. **At the** `ftp>` **prompt, type** `put` *filename remote_filename* **and press Return.**
5. **Type** `quit` **and press Return.**

▼ How to Copy a File From a Remote System

1. **Type** `ftp` *remote_system_name* **and press Return.**
2. **Type the user name for your account on the remote system and press Return.**

3. **Type the password for your account on the remote system (if required) and press Return.**
4. **At the** `ftp>` **prompt, type** `get` *filename local_filename* **and press Return.**
5. **Type** `quit` **and press Return.**

Note – The `ftp` command does not duplicate file permissions when copying files between systems. After you have copied the files, you may need to use the `chmod` command to reset the file permissions.

The following example shows how to copy a file from the remote system (`jupiter`) to the local system (`mars`):

```
% ftp jupiter
220 jupiter FTP server (UNIX(r) System V Release 4.0) ready
Name (mars:john):
331 Password required for john.
Password:
230 User john logged in.
ftp>get ~henry/test.c
200 PORT command successful.
150 ASCII data connection for /home/henry/test.c (IP address)
(20480 bytes).
226 ASCII Transfer complete.
local: /home/john/test.c remote: ~henry/test.c
21428 bytes received in 0.27 seconds (78 Kbytes/s)
ftp> quit
%
```

7

Enabling and Using Crash Dumps 8

This chapter contains these sections:

A system may *crash* or *hang* so that it no longer responds to commands. You can set up a system so that it automatically saves an image of the kernel when the system crashes. This image is called a *crash dump*. You can use the information in the crash dump files to diagnose and troubleshoot the cause of the failure.

This chapter describes how to enable crash dumps, and how to use the files and messages to help determine the cause of the failure.

What Happens When a System Crashes

When a system crashes, it:

- Aborts all running processes
- Tries to save recent data changes
- Displays an error message telling why it crashed

- Tries to write out a *crash dump* to the disk
- Tries to reboot the system

During operation, the system stores data in memory buffers, writing data to the disk only when necessary. If a system crashes, data stored in the buffers can be lost. To keep the system up to date, the operating system synchronizes the file system every 30 seconds. It runs the `sync` command, which updates the superblock and writes out any new information to the disk.

When a system crashes, data stored in memory may not have been completely written to disk. This can cause inconsistencies in file systems that must be repaired using `fsck`. See *File System Administration* and the `fsck(1M)` manual page for more information on `fsck`.

Error Messages Created by a Crash

When a system crashes, it displays a message like this:

```
panic: error message
```

where *error message* is one of the panic error messages described in the `crash(1M)` manual page.

Less frequently, this message may be displayed instead of the panic message:

```
Watchdog reset !
```

Crash messages are automatically stored in the `/var/adm/messages` file throughout the session. These messages are saved whether or not crash dumps are enabled for a system.

▼ How to Display Messages in `/var/adm/messages`

- **Type** `dmesg` **and press Return.**
  The contents of `/var/adm/messages` are displayed on the screen.

  or

- **Type** `more /var/adm/messages` **and press Return.**
  The contents of `/var/adm/messages` are displayed on the screen

See *Administration Supplement for Solaris Platforms* for examples of the `/var/adm/messages` file and a description of the booting messages.

What Is a Crash Dump?

A crash dump is the image of the state of the kernel that was in physical memory when the system failed. The physical memory (or `core` file) is a snapshot of the kernel containing all of the program text, data, and control structures that are part of the operating system. When a system crashes, the physical memory is written to the end of the swap slice of the disk. Although the system writes a `core` file whenever it crashes, it does not save the crash dump file unless you configure the system to do so.

How Crash Dumps Are Created

With crash dumps enabled, when you reboot a system after a crash, the `savecore` program runs. `savecore` preserves a copy of the crash dump by writing it from the end of the swap slice into the directory `/var/crash/`*systemname*, where *systemname* is the name of the system. `savecore` incrementally saves the core image in the file `vmcore.`*n* and the namelist for the kernel in the file, `unix.`*n*. The *n* suffix is incremented each time `savecore` is run. As a result, the `/var/crash/`*systemname* directory can grow quite large on a system that crashes repeatedly.

Before `savecore` writes out a `core` image, it tries to determine the amount of available space left in the file system by reading the `minfree` file in the `/var/crash/`*systemname* directory. The `minfree` file contains a single ASCII number that represents the number of kilobytes of free space that must remain available in the file system. If saving the `core` file reduces the minimum free space to below the number in the `/var/crash/`*systemname*`/minfree` file, then `savecore` does not write out the crash dump. If the `minfree` file does not exist, `savecore` always writes out the `core` file, if one was created.

One way you can control the size of the `/var/crash/`*systemname* directory is to edit the `minfree` file and set the number large enough to prevent `savecore` from writing out the `core` file.

8

You can save a crash dump manually on a system with crash dumps disabled by running `savecore` as soon as the system has completed booting. If you do not run `savecore` immediately, the swap space containing the crash dump will be overwritten by programs. See the `savecore(1M)` manual page for more information.

Enabling and Disabling Crash Dumps

Crash dumps are not enabled by default. Using the crash dump output requires detailed knowledge of the kernel and how it works. You should enable crash dumps only on individual systems that are experiencing frequent system crashes. Once the problem is diagnosed and fixed, disable crash dumps for that system. In other words, do not enable crash dumps unless you plan to use them.

To enable crash dumps, you must modify the `/etc/init.d/sysetup` file for the system.

▼ How to Create a Directory for Saving the `core` File:

1. **Become superuser.**
2. **Type** `cd /var` **and press Return.**
3. **Type** `mkdir crash` **and press Return.**
   The `/var/crash` directory is created.
4. **Type** `cd crash` **and press Return.**
5. **Type** `mkdir` *system-name* **and press Return.**
   A directory with the name of the system is created.

▼ How to Enable a Crash Dump

1. **Type** `vi /etc/init.d/sysetup` **and press Return.**
2. **Uncomment the lines that enable the crash dumps by deleting the comment marks (#) from the beginning of those lines.**
3. **Save the changes.**

This example shows the appropriate section of the /etc/init.d/sysetup file edited to enable crash dumps:

```
##
## Default is to not do a savecore
##
If [ ! -d /var/crash/‘uname -n‘ ]
then mkdir -p /var/crash/‘uname -n‘
fi
     echo ’checking for crash dump...\c ’
savecore /var/crash/‘uname -n`
     echo ’’
```

▼ How to Reserve File System Space

1. **Type** cd /var/crash/*system-name* **and press Return.**
2. **Create a file named** minfree **and specify the minimum available free space (in kilobytes) that must remain available in the file system.**
   For example, to reserve 5000 Kbytes of available free space, create a minfree file that looks like this:

```
saturn% more /var/crash/saturn/minfree
5000
saturn%
```

▼ How to Disable Crash Dumps

1. **Become superuser.**
2. **Edit the** /etc/init.d/sysetup **file**

3. **Insert a hash mark (#) at the beginning of each of the lines shown below:**

```
#if [ ! -d /var/crash/‘uname -n‘ ]
#then mkdir -p /var/crash/‘uname -n‘
#fi
#                echo ‘checking for crash dump...\c ‘
#savecore /var/crash/‘uname -n‘
#                echo ''
```

4. **Save the changes.**

5. **Type** `rm -rf /var/crash/`*system-name* **and press Return.**

Recovering From a Crash

This section describes how to recover from a crash, what to do if rebooting fails, and how to force a crash dump.

When a system crashes, you need to bring it back up before you can look at the crash dump files. After a crash, the system may reboot automatically.

What to Do if a System Hangs

If a system hangs, use this checklist:

- Make sure the pointer is in the window where you are typing the commands.
- Press Control-q in case the user accidently pressed Control-s, which freezes the screen. Note that, in a windowing environment, Control-s freezes only the window, not the entire screen. If a window is frozen, try using another window.
- Press Control-`\` to force a “quit” in the running program and (probably) write out a `core` file.
- Press Control-c to interrupt the program that may be running.
- If possible, log onto the system from another terminal or remote login from another system on the network. Type `ps -ef` and look for the hung process. If it looks like the window system is hung, find the process and kill it.

- Try becoming superuser and rebooting the system.
- If the system still does not respond, force a crash dump and reboot. See *Administration Supplement for Solaris Platforms* for information on forcing a crash dump and booting.
- If the system still does not respond, turn the power off, wait a minute or so, then turn the power back on. This procedure is frequently called *power cycling*.
- If you cannot get the system to respond at all, contact your local service provider for help.

What to Do if Rebooting Fails

After a crash, the system may reboot automatically. If the automatic reboot fails with a message such as:

```
reboot failed: help
```

then run `fsck` in single-user mode.

If the system does not reboot, or if it reboots and then crashes again, there may be a hardware problem with a disk or one of the boards.

Check your hardware connections:

- Make sure the equipment is plugged in.
- Make sure all the switches are in the proper settings and pushed all the way in.
- 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 you cannot find any obvious fault with the connections, and the system still refuses to respond, contact your local service provider.

Before You Call for Help

Before calling for help, make sure you have accurately copied down crash messages from the console or taken them from the `/var/adm/messages` files.

If you are having frequent crashes, gather all the information you can about them and have it ready when you call for help.

Using a Crash Dump

Use the `crash` kernel debugger to examine the memory images of a live or crashed system kernel. You can examine the control structures, active tables, and other information about the operation of the kernel. The syntax of the command is:

`/usr/sbin/crash [ –d` *dump-file* `] [ –n` *name-list* `] [ –w` *output-file* `]`

Only a few aspects of `crash` are useful to a system administrator. Completely describing the crash debugger is beyond the scope of this book. To use `crash` to its full potential requires a detailed knowledge of the kernel. Saved crash dumps can, however, be useful to send to a customer service representative for analysis. For details on the operation of the crash utility, see the `crash(1M)` manual page.

Additional Diagnostic Techniques

Log files and system messages provide information that can help determine what is wrong with a system that hangs, crashes, or does not reboot. This section describes how to read and use these messages.

Looking at Messages Generated During Booting

The `/usr/sbin/dmesg` command displays the error messages generated during booting. You can view these messages or redirect them to a file.

See *Administration Supplement for Solaris Platforms* for alternate methods of displaying boot messages.

Using System Error Logging (`syslogd`*)*

Many system facilities use the error logging daemon, `syslogd`, to record messages whenever an unusual event occurs. Typically, these messages are written to `/var/adm/messages` or to the system console. These messages can help you determine the cause of problems with a system. For example, an increasing number of error messages coming from a device may be an indication that the device is about to fail.

Setting Up System Logging

To set up system logging, you must have an `/etc/syslog.conf` file. This file has two columns: the first column specifies the source of the error condition and its priority; the second specifies the place where the errors are logged.

The message sources are specified by two parts separated by a dot (`.`). The first part is the source or *facility*, which describes the part of the system generating the message. The second part is the priority of the message. The most common sources are shown in Table 8-1. The most common priorities are shown in Table 8-2.

Table 8-1 Sources for `syslog.conf` Messages

Source	Meaning
`kern`	The kernel
`auth`	Authentication
`daemon`	All daemons
`mail`	Mail system
`lp`	Spooling system
`user`	User processes

Note – There is a maximum of 24 syslog sources (or facilities) that can be activated in the `/etc/syslog.conf` file.

Table 8-2 Priorities for `syslog.conf` Messages

Priority	Meaning
`err`	All error output
`debug`	Debugging output
`notice`	Routine output
`crit`	Critical errors
`emerg`	System emergencies
`none`	Don't log output

For, example, the entries:

```
user.err                    /dev/console
user.err                    /var/adm/messages
mail.debug                  /var/log/syslog
```

show that user errors are printed to the console and are also logged to the file `/var/adm/messages`. Mail debugging output is logged to the file `/var/log/syslog`.

The following example shows the default `/etc/syslog.conf` file:

```
#ident  "%Z%%M% %I%     %E% SMI"        /* SunOS 5.0 */
#
# Copyright (c) 1991-1993, by Sun Microsystems, Inc.
#
# syslog configuration file.
#
# This file is processed by m4 so be careful to quote (‘’) names
# that match m4 reserved words.  Also, within ifdef's, arguments
# containing commas must be quoted.
#
# Note: Have to exclude user from most lines so that user.alert
#       and user.emerg are not included, because old sendmails
#       will generate them for debugging information.  If you
#       have no 4.2BSD based systems doing network logging, you
#       can remove all the special cases for "user" logging.
#
```

```
*.err;kern.notice;auth.notice;user.none          /dev/console
*.err;kern.debug;daemon,auth.notice;mail.crit;user.none /var/adm/messages

*.alert;kern.err;daemon.err;user.none           operator
*.alert;user.none                               root

*.emerg;user.none                               *

# if a non-loghost machine chooses to have authentication messages
# sent to the loghost machine, un-comment out the following line:
#auth.notice                   ifdef(‘LOGHOST’, /var/log/authlog, @loghost)

mail.debug                     ifdef(‘LOGHOST’, /var/log/syslog, @loghost)

#
# non-loghost machines will use the following lines to cause "user"
# log messages to be logged locally.
#
ifdef(‘LOGHOST’, ,
user.err                                        /dev/console
user.err                                        /var/adm/messages
user.alert                                      'root, operator'
user.emerg                                      *
)
```

The `/var/adm` directory contains several message files. The most recent messages are in `/var/adm/messages` (and in `messages.0`), and the oldest are in `messages.3`. After a period of time (usually every ten days), a new messages file is created. The file `messages.0` is renamed `messages.1`, `messages.1` is renamed `messages.2`, and `messages.2` is renamed `messages.3`. The current `/var/adm/messages.3` is deleted.

8

Index

S

U

V

W