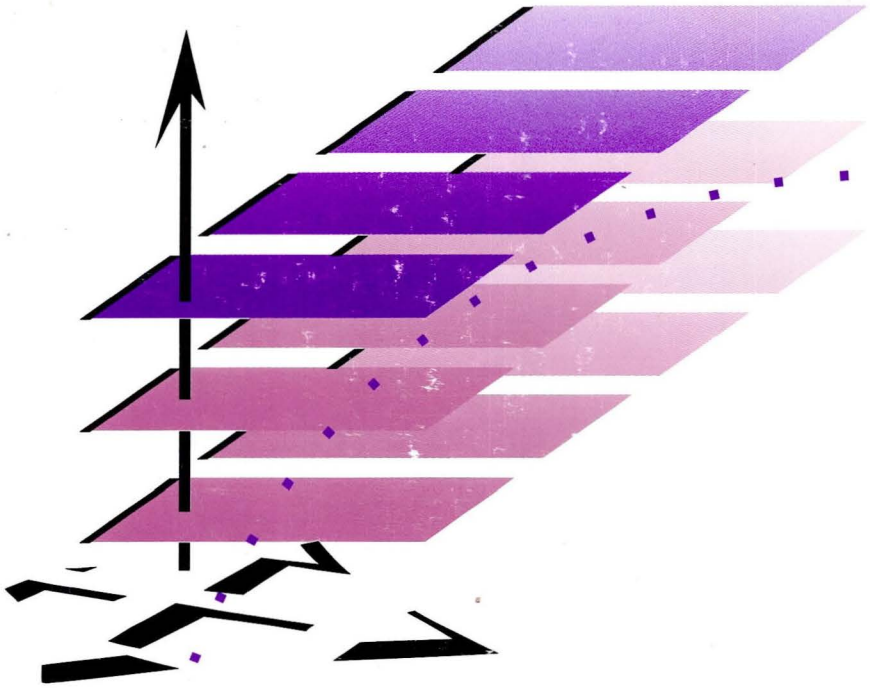


CONVEX

SPP-UX System Administration Guide

First Edition

Convex Computer Corporation
3000 Waterview Parkway
P.O. Box 833851
Richardson, TX 75083-3851
United States of America
(214) 497-4000

SPP-UX System Administration Guide

Order No. DSW-853

October 1994

Convex Press
Richardson, Texas
United States of America

SPP-UX
System Administration Guide

Order No. DSW-853

Printed in the United States of America

Revision information for SPP-UX System Administration Guide

Edition	Document No.	Description
First	710-029330-000	Initial release October, 1994.

Contents

Preface xiii

Audience xiii
How this book is organized xiv
Notational conventions xiv
General conventions xiv
Command syntax xv
Associated documentation xvi
Ordering documents xvi
Technical assistance xvii

1. Introduction 1

Setting up an SPP-UX system 1
The system administrator's interface to SPP-UX 1
Accessing the SPP-UX system console remotely 2
Disabling remote access to the SPP-UX system console .. 2
Accessing system administration utilities 3
The system administrator's responsibilities 3
Using SAM 4
SAM Interfaces 4
Starting SAM 4
Quitting SAM 5
A sample SAM session: Creating a new user account 6
Getting help in SAM 8

2. Configuring an SPP-UX system 9

Installing and upgrading SPP-UX 9
Installing and upgrading optional software 9
Post-installation tasks 10
Common post-installation tasks 10
First-time post-installation tasks 11
Setting system parameters 11
Adding the system to a network 12
Selecting a host name for your system 12
Getting an Internet address 12
Setting your system's time zone 12
Setting your system clock 13
Setting the root password 14
Setting up networking 14
Setting up man pages 15

Adding users and groups 16
Setting up electronic mail 16
Setting up the line-printer spooler 17
Setting up news 18
Setting up system accounting 19
Customizing the system 19
Customizing system startup 20
Customizing the login prompt (etc/issue) 21
Posting a message of the day (/etc/motd) 22
Customizing users' login environments 22
Customizing users' editing environments 23
Setting up non-standard terminal types 23

3. The Subcomplex Manager utility 27

Subcomplex Manager interfaces 28
The Subcomplex Manager graphical user interface 28
Viewing information about system resources 29
Creating and saving a complex configuration file 29
Loading a complex configuration file into the Subcomplex Manager 30
Overlaying a complex configuration file into the Subcomplex Manager 30
Performing a system reconfiguration 30
Subcomplex Manager windows 31
Subcomplex Manager main window 31
Node Attributes dialog box 34
Subcomplex Attributes dialog box 35
Reconfiguration constraints 36
Error messages 38
The Subcomplex Manager command-line interface 38
scm command options 38
scm command examples 39
Subcomplex manager configuration file format 40
Example SCM configuration file 42

4. Starting and stopping SPP-UX 43

Turning on the system 43
Starting SPP-UX 44
Reviewing the state of the file system 45
Shutting down the system 46
An overview of shutdown 47
Going to the single-user mode for maintenance 47
Shutting down the system completely 48
Designating system shutdown authorization 49
Customizing the shutdown process 50
Power failure considerations 50
Power failure-related tasks 50

Using SAM to halt or reboot the system 50

5. Controlling access to SPP-UX..... 53

Terms used in this chapter 53
Overview of controlling access your system 55
Controlling user accounts and groups 55
Controlling file access 59
Controlling run-levels 62
Managing user accounts and group tasks 63
Adding a user using SAM 65
Additional task information 65
Removing a user using SAM 67
Additional Task Information 67
Customizing the SAM 'Adding and Removing a User' capabilities 68
Additional task information 69
Deactivating a user's account using SAM 71
Additional task information 71
Reactivating a user's account Using SAM 72
Additional task information 73
Displaying/modifying a user's account information using SAM 73
Additional Task Information 74
Adding a group using SAM 74
Additional Task Information 75
Removing a group using SAM 76
Additional task information 76
Adding and removing users from groups using SAM 77
Additional Task Information 77
Adding a user using SPP-UX commands 78
Additional task information 84
Examples 86
Removing a user using SPP-UX commands 88
Additional task tnformation 90
Examples 90
Deactivating a user's account using SPP-UX commands ... 92
Additional task information 93
Examples 93
Reactivating a user's account using SPP-UX commands 94
Additional Task Information 95
Displaying/modifying a user's account information using SPP-UX commands 95
Additional task information 97
Examples 97
Adding a group using SPP-UX commands 98
Additional task information 99
Examples 100
Removing a group using SPP-UX commands 101

Additional task information 102
Changing a user's primary group using SPP-UX commands . 103
Additional task information 104
Adding users to groups using SPP-UX commands 104
Additional task information 105
Examples 106
Removing users from groups using SPP-UX commands .. 106
Additional task information 108
Displaying/assigning special group privileges using SPP-UX commands 108
Additional task information 110
Examples 111
Managing run-levels 111
Creating a new run-level using SPP-UX commands 112
Examples 113
Changing system run-levels using SPP-UX commands 113
Entering the system administration run-level 114
Returning from the system administration run-level 115

6. Managing an SPP-UX file system 117

Terms used in this chapter 117
Overview of SPP-UX file systems 120
What does "file system" mean? 120
Mountable file systems 120
Listing mounted file systems 121
Types of file systems 121
Disk layout 122
SPP-UX system files 122
The diskutil disk utility 125

7. Managing printers 131

What is the Line Printer Spooling System? 131
Terms used in this chapter 132
LPspooler overview 135
The components of the LP spooler 136
Remote spooling 142
Priorities of printers and print requests 143
Using plotters with the spooler 144
Controlling data flow* through the spooler 145
Logging and analyzing printer activity 146
Initial spooler setup 147
Spooler tasks 147
Additional task information 148
Viewing printers and print request status using SAM 149
Additional task information 149
Adding a remote printer using SAM 150

Additional task information 152
Adding a network-based printer using SAM 153
Additional task information 154
Removing a printer using SAM 155
Additional task information 155
Starting and stopping the spooler using SAM 156
Additional task information 157
Determining the status of the spooler using SAM 158
Additional task information 158
Disabling a printer using SAM 159
Additional task information 159
Enabling a printer using SAM 160
Additional task information 160
Changing a printer fence priority using SAM 161
Additional task information 162
Setting up the spooler using SPP-UX commands 162
Determining if a device file exists for your printer using SPP-UX commands 163
Additional task information 163
Adding a remote printer using SPP-UX commands 164
Additional task information 167
Examples 167
Adding a network-based printer using SPP-UX commands ... 169
Additional task information 169
Creating a printer class using SPP-UX commands 169
Additional task information 170
Examples 170
Removing a printer or printer class using SPP-UX commands 171
Additional task information 172
Examples 173
Accepting and rejecting print requests for a printer using SPP-UX commands 173
Additional task information 174
Examples 174
Enabling or disabling a printer using SPP-UX commands 175
Additional task information 175
Examples 176
Setting a printer fence priority using SPP-UX commands 176
Additional task information 177
Starting and stopping the spooler using SPP-UX commands . 177
Additional task information 177
Examples 177
Canceling print requests using SPP-UX commands 178
Additional task information 178
Examples 179

Moving all requests using SPP-UX commands 179
Examples 181
Moving selected print requests using SPP-UX commands 181
Additional task information 182
Examples 182
Viewing the status of printers and print requests using SPP-UX commands 182
Additional task information 183
Examples 184
Changing the priority of print requests using SPP-UX commands 185
Additional task information 185
Examples 186
Displaying statistics about printer activity using SPP-UX commands 186
Additional task information 187
Examples 187

8. System accounting 189

Installation and daily usage 189
How to install System Accounting 190
Summary of daily operation 192
Overview of System Accounting 193
Definitions 193
Introduction to commands 195
Login and directory structure 199
Disk space usage accounting 200
Reporting disk space usage 200
Creating total accounting records 204
Connect session sccounting 205
Writing records to wtmp (acctwtmp) 206
Displaying connect ression records (fwtmp) 206
Creating Total Accounting Records 209
Process accounting 213
Turning process accounting on 213
Turning process accounting off 214
Checking the size of pacct 215
Displaying process accounting records using acctcom 217
Command summary report (acctcms) 224
Creating total accounting records 229
Charging fees to users (chargefee) 230
Summarizing and reporting accounting information 231
Displaying total accounting records (prtacct) 232
Merging total accounting files (acctmerg) 234
Creating daily accounting information (runacct) 236
Displaying runacct reports (prdaily) 242
Creating monthly accounting reports (monacct) 245
Updating the holidays file 246

Fixing corrupted files 247
Fixing wtmp errors 247
Fixing tacct errors 247
Sample accounting shell scripts 248
grpdusg 248
System Accounting files 253
Files in the /usr/adm directory 253
Files in the /usr/adm/acct/nite directory 253
Files in the /usr/adm/acct/sum directory 254
Files in the /usr/adm/acct/fiscal directory 255

A SPP-UX system tunable parameters .257

SPP-UX system tunables file 257

B Crashdump.....265

Oveview 265
Creating a crashdump partition 265
Creating a crashdump file(crashutil) 266

Preface

The *SPP-UX System Administration Guide* describes the fundamental concepts and tasks associated with setting up and maintaining an SPP-UX system.

Audience

This manual is intended for system administrators of all skill levels. However, there are basic skills that you need to have before attempting to administer an SPP-UX system. If you have acquired these skills by using a Unix operating system, or a Unix-related operating system such as ConvexOS or HP-UX, you will be able to perform these tasks with SPP-UX.

You should be able to perform the following tasks:

- Log in and out of a remote computer system (using a command such as `rlogin` or `telnet`)
- Understand how the SPP-UX file system works, how to navigate through the file system using the `cd` command, and how to use both relative and absolute path names
- Understand the SPP-UX file permission system, and be able to change permissions using the `chmod`, `chgrp`, and `chown` commands
- Edit files using one of the SPP-UX editors (such as `vi` or `emacs`)
- Move, copy, and delete files using `mv`, `cp`, and `rm`, respectively
- Search for text in files using `grep`
- Display the contents of files using the `head`, `cat`, and `more` commands
- Use one of the SPP-UX shells (`csh`, `ksh`, or `sh`)

- Understand how SPP-UX processes work, and know how to start, stop, and display processes
- Run an X Window System server on your local host

How this book is organized

This manual is divided into the following sections:

- Chapter 1 presents an overview of SPP-UX system administration and the primary interface for system administration, SAM
- Chapter 2 describes how to set up your system following the installation of your computer
- Chapter 3 describes the Subcomplex Manager (SCM) utility
- Chapter 4 describes how to start up (boot) and shut down your system
- Chapter 5 describes how to create user accounts, user groups, and file access permissions
- Chapter 6 describes how to manage the SPP-UX file system
- Chapter 7 describes how to manage printer output
- Chapter 8 describes system accounting concepts and procedures
- Appendix A describes the system tunable parameters

Notational conventions

This section discusses notational conventions used in this book.

General conventions

In general, the following conventions are used in this guide:

- *Italic*
  - Designates user-supplied variables in a command-line or code example

  - Introduces new and important terms
  - Identifies variables in mathematical equations
  - Indicates document titles
- `Constant-width font` designates input and output, including
  - Command names and options
  - System calls
  - Data structures and types
  - Variables and arrays
  - Function and subroutine names
  - Directives, program statements, display examples, printout examples, and error messages returned
- Horizontal ellipsis (...) shows repetition of the preceding item(s).
- Vertical ellipsis shows that lines of code have been left out of an example.

References to man pages appear in the form mnpgname(1), where "mnpgname" is the name of the man page and is followed by its section number enclosed in parentheses. To view this man page, you would type:

```
man 1 mnpgname
```

Note

A Note highlights important supplemental information.

Caution

A Caution highlights procedures or information necessary to avoid damage to equipment, software, or data.

Command syntax

Consider this example:

```
COMMAND input_file [...] {a | b}
[output_file]
```

1. COMMAND must be typed as it appears.

2. *input_file* indicates a file name that must be supplied by the user.
3. The horizontal ellipsis in brackets indicates that additional input file names may be supplied.
4. Either a or b must be supplied.
5. [*output_file*] indicates an optional file name.

Associated documentation

- For information on the Exemplar architecture, refer to the *Exemplar Architecture* manual (DHW-014).
- For information on installing and ugrading SPP-UX and software products that run under SPP-UX, refer to the *Exemplar Software Installation* manual (DSW-852).
- For more information on using SPP-UX, refer to the *HP-UX Reference* (Hewlett-Packard order number B2355-90004).

Ordering documents

To order the current edition of this or any other Convex document, send requests to:

Convex Computer Corporation
Customer Service
P.O. Box 833851
Richardson TX 75083-3851 USA

Please include the order number (DSW or DHW number) or the exact title of the document.

Technical assistance

If you have questions that are not answered in this book, contact the Convex Technical Assistance Center (TAC) at the following locations:

- Within the continental U.S., call 1 (800) 952-0379.
- From Canada, call 1 (800) 345-2384.
- All other locations, contact the local Convex office.

You can also use the `contact` utility to report any problems you may have with SPP-UX or its associated documentation. For more information, refer to the contact(1) man page on any Convex computer system.

Introduction 1

Setting up an SPP-UX system

Each Exemplar computer system is shipped with a bootable copy of the most recent release of SPP-UX. You should first read and follow the instructions in the copy of the *SPP-UX Release Notice* that you receive with your release of SPP-UX. When you complete the instructions presented in the Release Notice, you should continue by configuring your system as described in Chapter 2, "Configuring an SPP-UX system," on page 9.

When you receive a new version of SPP-UX, read and follow the instructions in the copy of the *SPP-UX Release Notice* you receive with the new release of SPP-UX. You will need to follow the installation procedures described in the *SPP-UX Installation and Upgrade* manual to install the new release of SPP-UX. When you complete the installation procedures, continue by configuring your system as described in Chapter 2, "Configuring an SPP-UX system," on page 9.

The system administrator's interface to SPP-UX

The SPP-UX system console starts up automatically on the Exemplar test station when SPP-UX is booted on the Exemplar system. The `sn.cnsld` daemon must be running on the test station in order for the SPP-UX system console to run; if `sn.cnsld` is not running, you can start it by entering the following command on the test station (you must be logged in to the test station as root):

```
/spp/etc/sn.cnsld
```

The SPP-UX system console appears in an xterm window with a title bar that reads "`System Console`."

Accessing the SPP-UX system console remotely

You can access the SPP-UX system console remotely (that is, from a system other than the test station) by using the `sn_cnsl` command on the Exemplar test station. You must be logged in to the test station (either locally or remotely) as root in order to use this command.

The `sn_cnsl` command allows you to either remotely view the SPP-UX console window, or to remotely assume control of it. The `sn_cnsl` command has the following syntax:

```
sn_cnsl -f | -s
```

The `sn_cnsl` command supports the following options:

`-f`	Force control of the SPP-UX system console to the host from which the command was entered
`-s`	Spy on the main SPP-UX system console; all system messages written to the SPP-UX system console are copied to the host from which the command was entered

Disabling remote access to the SPP-UX system console

If, for security reasons, your site does not want to allow remote access to the SPP-UX system console, you can disable remote access. To disable remote access, change the permissions for the `sn_cnsl` command (or remove it from the test station). The following command removes execute permission from the `sn_cnsl` command:

```
chmod 600 /spp/bin/sn_cnsl
```

Accessing system administration utilities

You can access the SPP-UX system administration utilities from any host that is connected to your Exemplar system. Most system administration utilities require that you be logged in to the Exemplar system as `root`; some require you to be logged in as `adm`.

The SAM and SCM graphical user interfaces require that your local host be running an X Windows server. The SPP-UX X client is compatible with X11R4 and later versions of X Windows; an X11R5 server on your local host is recommended.

The system administrator's responsibilities

The SPP-UX system administrator is responsible for installing and configuring the SPP-UX operating system software, and for maintaining the system and repairing it when something goes wrong. More specifically, the SPP-UX system administrator may need to do the following things:

- Install the SPP-UX operating system software
- Install software products that run under SPP-UX
- Configure the SPP-UX operating system
- Update the SPP-UX operating system software
- Create and maintain user login accounts for the system
- Configure, and manage peripheral devices on the system
- Monitor file system use and growth
- Back up and restore files
- Detect and correct file system errors
- Assist others in using the system
- Provide a backup system administrator to assist users (when the primary administrator is unavailable)

Using SAM

SAM is the System Administration Manager utility for SPP-UX. This tool allows you to perform many system administration tasks without using the underlying SPP-UX commands that are associated with the task. SAM can save you time and keystrokes.

SAM can help you with tasks in the following areas:

- Working with users' accounts
- Working with groups of users
- Maintaining system security
- Working with file systems
- Configuring your swap space
- Adding or removing peripherals
- Working with the line printer spooler
- Backing up and recovering files (automated or manual system backups)
- Configuring network connections (LAN, X.25, FDDI, and TokenRing)
- Configuring UUCP communication
- Administering systems remotely from one location

SAM Interfaces

SAM has two user interfaces, an X Window System graphical user interface (GUI) and a text terminal interface. The two interfaces differ in the screen appearance and keyboard/mouse interactions. If you have and X server available, it is strongly recommended that you use the GUI. The examples in this manual will generally show how to perform tasks using the GUI.

Starting SAM

To start up SAM, first make sure that you are logged in to the Exemplar system with superuser (root) privileges. Since you will be running the SAM GUI

on your local host, make sure that your DISPLAY environment variable is set to your local host; if your are running csh or tcsh, you would enter the following command:

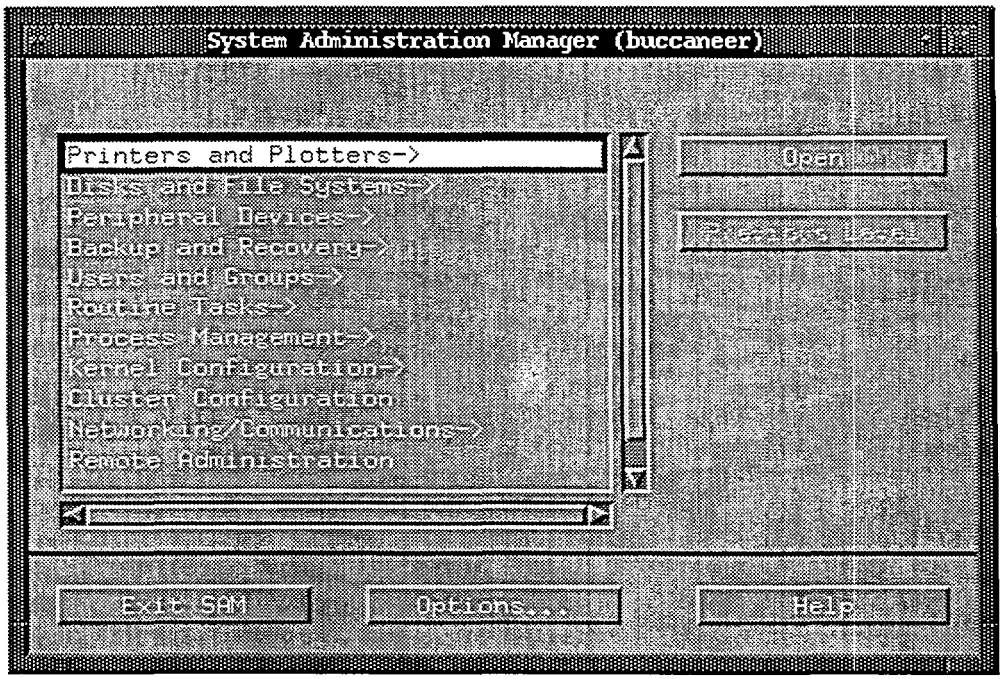
```
setenv DISPLAY localhost:0.0
```

Start SAM by entering the following command:

```
/usr/bin/sam
```

The SAM main menu screen will appear, which will appear as shown in Figure 1 in an xterm window:

Figure 1 SAM main window



Quitting SAM

To exit SAM from the SAM main window, press the `Exit SAM` button in the lower left corner of the window. If you are in a different SAM window, first select the `Exit` option from the `List` menu in that window to get to the SAM main window.

Note

If a popup window or error message box is displayed, you will have to first close the window or box before you can perform actions in the SAM windows.

A sample SAM session: Creating a new user account

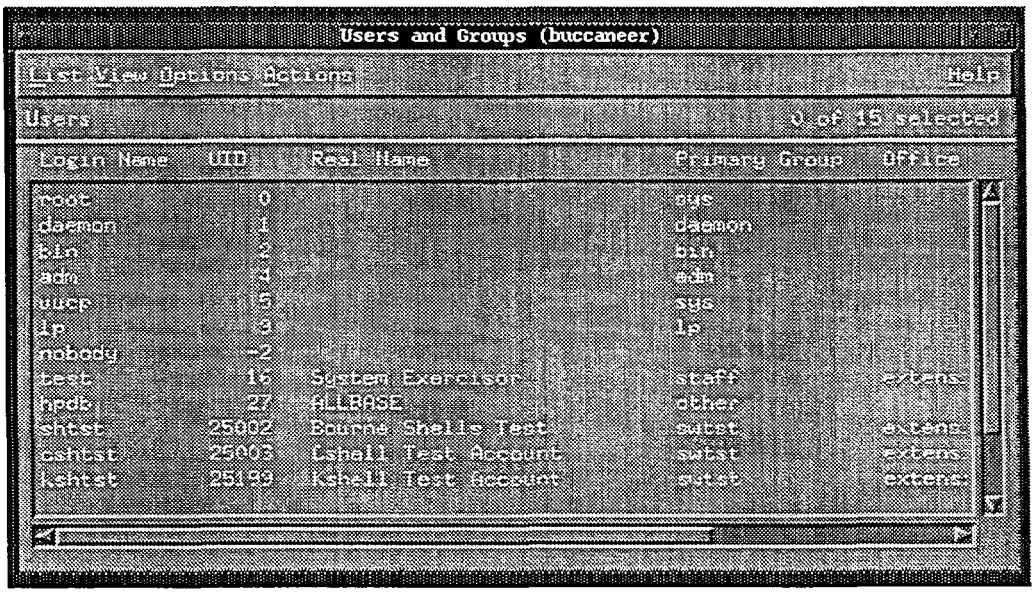
Step 1 To add a new user to the system, first log in to your Exemplar system and start a SAM session:

```
rlogin exemplar -l root
setenv DISPLAY myhost:0.0
/usr/bin/sam
```

Step 2 Select `Users and Groups` from the SAM main window. You can do this either by left-clicking on the `Users and Groups` option and then left-clicking on the `Open` button, or by double-left-clicking on the `Users and Groups` option.

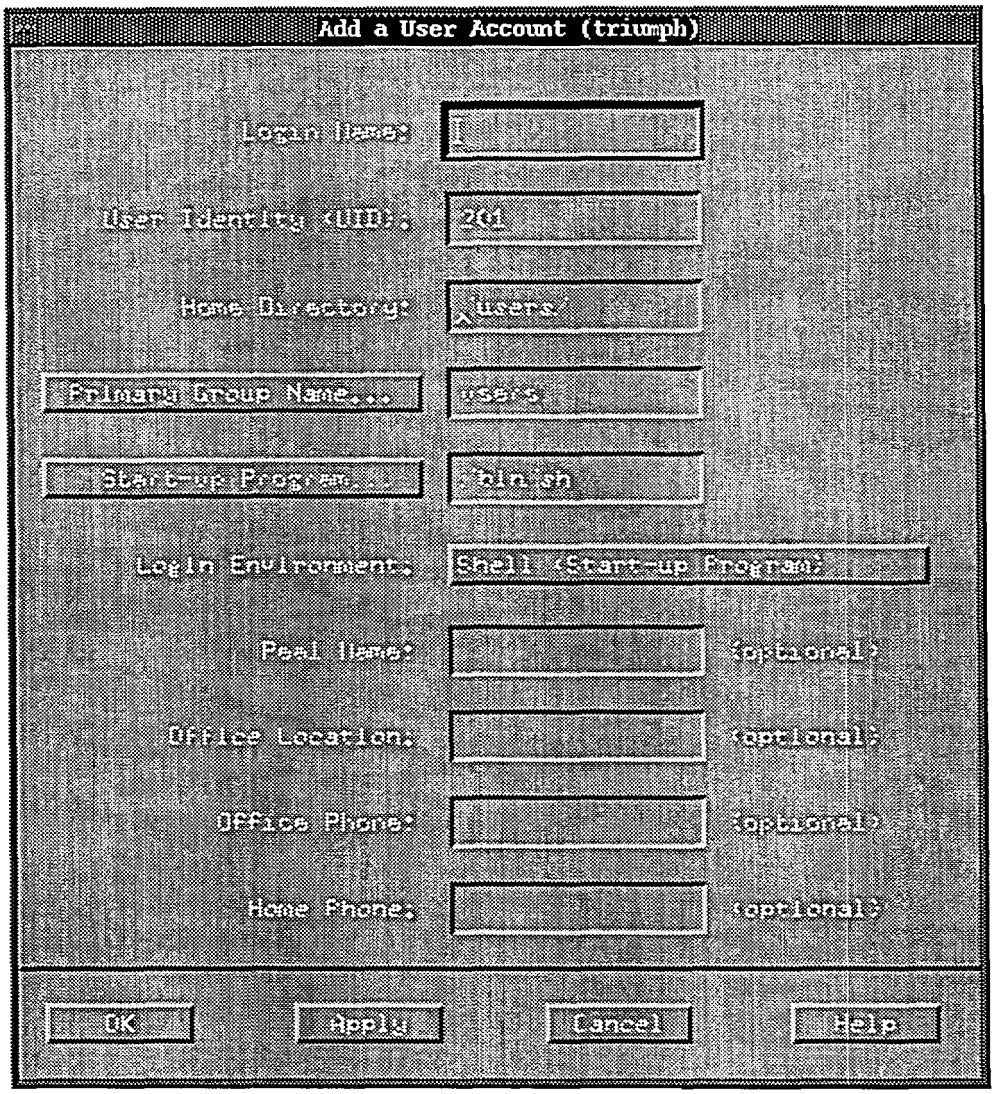
Selecting the `Users and Groups` option displays the Users and Groups window, as shown in Figure 2:

Figure 2 SAM Users and Groups window



Step 3 To add a new user, select the Add option from the Actions menu. The Add a User Account window appears, as shown in Figure 3:

Figure 3 SAM Add a User Account window



Some of the data entry fields have buttons either to the left of the field or in the field itself. Pressing the button allows you to choose from a list of options for the data entry field. If the button is in the data entry field, as is the case with the Login Environment field in the Users and Groups window, you may only choose one of the items in the list shown when you press the button. If the button is to the left of the data

entry field, as is the case with the Primary User Name and Start-up Program fields, you can either select one of the items from the button's list or enter a different value.

Step 4 All the required fields except the Login Name field have default information already supplied. Enter a login name, change any of the default values you wish to change, enter information into the optional fields if you wish, then press the Apply button to create a new account.

Step 5 A popup window will ask you to enter a password for the new account. Enter a password for the new user (or press the OK button to create the account with no initial password); you will be prompted to reenter the password to verify it, then press OK to create the account. A popup message window will verify that the new account has been created; press the OK button to close this window.

The new user account that you created will appear immediately in the SAM Users and Groups window.

Step 6 Quit your SAM session by selecting Exit from the List menu of the Users and Groups window, then press the Exit SAM button in the SAM main window.

Getting help in SAM

The SAM GUI contains extensive on-line help. All windows, with the exception of message boxes and some dialog boxes, have on-line help available. If the window has a menu bar along the top, select the Help menu (the rightmost menu) from the menu bar. If the window has action buttons but no menu bar, press the Help button.

Configuring an SPP-UX system

2

This chapter explains how to build a new SPP-UX system. It outlines what you need to do to immediately after installation to create a functioning system, and tells you where you can find more detailed information if you need it.

Installing and upgrading SPP-UX

As the system administrator, you may be responsible for installing SPP-UX. Your Exemplar system is shipped with a bootable copy of SPP-UX. There are two circumstances that will require you to install SPP-UX:

- If your working copy of SPP-UX is damaged or destroyed (for example, if the disk containing SPP-UX crashes), you will need to perform a scratch installation of the operating system
- When you receive a new version of SPP-UX, you will upgrade your operating system to the new version

The *Exemplar Software Installation* manual provides instructions for performing both an SPP-UX scratch installation and an SPP-UX upgrade.

Installing and upgrading optional software

You can use the installation procedures described in the *Exemplar Software Installation* manual to install or upgrade optional Convex software (such as compilers) to your system, or to update optional software products to a new release. These procedures can also be used to install software supplied by other vendors.

When adding and updating software, be sure to follow the supplier's directions. Make sure that you use the appropriate installation utility (SD, `update`, or `tar`) for the software you are installing.

Post-installation tasks

The following is a recommended set of system administration tasks that should be performed after installing or upgrading SPP-UX.

Common post-installation tasks

Action	Description
Run `set_parms`	Add the Exemplar system to a network, set the host name, set the system clock (time zone, date, and time), and set the Internet (IP) address for the system. See "Setting system parameters" on page 11.

Note

`set_parms` runs automatically after a new version of SPP-UX is installed. If you exit the `set_parms` interactive session prematurely (without providing all the information requested), your system will be automatically shut down and will require a power-on reboot. If you provide the requested information, the system files that use this information are updated immediately.

Set root password	Change the root (superuser) password from the value that is shipped with the new version of the system to the value you will use at your installation. See "Setting the root password" on page 14.
Update log file	Log in as root and check for problems in `/tmp/update.log`. Follow any instructions you find in this file.
Set up networking	Read the networking documentation supplied with SPP-UX and follow the instructions appropriate for the type of network at your installation. See "Setting up networking" on page 14.
Set up man pages	You can preformat man pages and remove the man page source from your system to save disk space. See "Setting up man pages" on page 15.
Install optional software	Read the release notices for any new or updated optional software products you wish to install on the new version of SPP-UX, and follow the

installation procedures in the *Exemplar Software Installation* manual to install them.

First-time post-installation tasks

If this is the first time you have installed SPP-UX on your system, you will need to perform the following tasks after installing SPP-UX.

Action	Description
Add users	Create login accounts for system users. See "Adding users and groups" on page 16.
Add user groups	Create user groups for system users. See "Adding users and groups" on page 16.
Create file systems	Mount file systems onto the root file system. See Chapter 6.
Set up electronic mail	Enable system users to send and receive electronic mail. See "Setting up electronic mail" on page 16.
Set up a printer spooler	Manage printers using the printer spooling system. See "Setting up the line-printer spooler" on page 17.
Set up news	Set up an electronic news system. See "Setting up news" on page 18.
Create an accounting system	Set up an accounting system appropriate for the needs of your installation. See "Setting up system accounting" on page 19.
Customize the system	There are a number of system features you can customize, including system startup, the system login prompt, and setting up non-standard terminal types. See "Customizing the system" on page 19.
Create a crashdump partition	Create a raw disk partition to hold crashdump data in the event of a system crash. See "Creating a crashdump partition" on page 265.

Setting system parameters

When you first log on to your SPP-UX system after installing or upgrading SPP-UX, the `/etc/set_parms` program automatically executes. This program adds your Exemplar system to a network, sets the host name, sets the system clock (time zone, date, and time), and sets the Internet (IP) address for the system.

Adding the system to a network

The `/etc/set_parms` program first asks if you are ready to link your system to a network. In order to complete this procedure, you will need to provide the following information:

- The host name for your system
- The Internet (IP) address for your system
- The time zone your system will use
- The system clock time

You will need to have this information ready when `/etc/set_parms` executes. If you do not, you will need to exit the program and run it again when you have the information.

Selecting a host name for your system

You can select any host name for your system containing up to eight characters. As a convention, you may want to select the same host name for your Exemplar system and your Exemplar test station. If you do this, you must select a base name containing no more than six characters; the characters `_d` are automatically appended to the test station's host name on the network connecting the test station to the Exemplar diagnostic bus, and the test station's host name on the external Ethernet port has the characters `_t` appended to the base host name.

Getting an Internet address

You will need to get an Internet address for your system from your network administrator. This address consists of four integer fields separated by periods. The first three fields are the address of your network's subnet, and the last field is the host address of your system on that subnet.

Setting your system's time zone

The `/etc/set_parms` program will set the time zone for your system based on the information you provide about your geographic location. You will first be prompted to specify your country, then your

state or province, and so on until /etc/set_parms has enough information to determine your time zone setting.

The value you select for your system's time zone is copied to the following system files:

- /etc/rc
- /etc/profile
- /etc/csh.login

Setting your system clock

The /etc/set_parms program will display the current system clock value (date and time), and allow you to change the setting. If you choose to change the clock setting, you will be promted separately to enter the month, day of the month, hour (in 24-hour format), minutes, and year.

Caution

Changing the system clock setting on a system can cause problems for some utilities. You should notify your system's users in advance if you change the system clock setting. The following are known problems:

make

The make program is sensitive to a file's time and date information and to the current value of the system clock.

Setting the clock forward will not affect make, but setting the clock backward by even a small amout may cause make to behave unexpectedly.

cron

Changing the system clock setting can cause unexpected results for jobs scheduled by cron:

- If you set the clock back, cron does not run any jobs until the clock catches up to the point from which it was set back

  For example, if you set the clock back from 8:00 to 7:00, cron will not run any jobs until the clock again reaches 8:00.

- If you set the clock ahead, cron attempts to catch up by immediately starting all jobs scheduled to run between the old time and the new time.

For example, if you set the clock ahead from 9:00 to 10:00, `cron` immediately starts all jobs scheduled to run between 9:00 and 10:00.

Setting the root password

Each new version of SPP-UX will be shipped with a default root password. This password will be documented in the SPP-UX Release Notice. For security reasons, you should immediately change the root password after installing SPP-UX. Use the `/bin/passwd` command to change the root password.

Caution

Don't forget the root password to your system! You will need to perform a system recovery if none of the system administrators can remember the root password.

Setting up networking

Your Exemplar system is shipped with an FDDI network adapter and associated driver software.

What you need to do to set up networking depends on the type of network you're using, and whether you're connecting the computer to an existing network or setting up a new one.

The following HP manuals cover the information necessary to install networking services on your Exemplar system. These manuals are shipped along with SPP-UX:

- *Administering and Installing FDDI*
- *Using NFS Services*
- *Administering and Installing NFS Services*
- *Programming and Protocols for NFS Services*
- *Using ARPA Services*
- *Installing and Administering ARPA Services*

Setting up man pages

Every SPP-UX command, system call, library function, and system file is documented in the SPP-UX man pages.

The man pages require some setup. You have three methods to choose from:

1. Create a formatted version of all the manpages.

   The advantage of doing this is that users will get quick response when they call up a man page on-line. The disadvantage is that the formatted versions take up a considerable amount of disk space (about as much again as the `nroff` originals from which they are created), which you may not have. However, once the pages have been formatted, you can recover disk space by getting rid of the `nroff` originals.

   This is a good method if you have enough disk space to hold both versions of the man pages for as long as it takes to finish formatting them.

   If you decide to use this method, enter the following command:

   ```
   /etc/catman
   ```

   The process of formatting all the man pages can take as long as five or six hours, so you might want to run it at a lower priority, in the background, or at night.

2. Format only certain sections of the man pages.

   This could give you the advantage of quicker access to heavily used sections without incurring the cost in disk space of formatting all sections.

   If you decide to use this option, enter the following command:

   ```
   /etc/catman sections
   ```

   where *sections* is one or more of the man page sections.

3. Do not execute `/etc/catman` at all.

   Use this method if you can spare some disk space but do not want to use any more than is necessary.

   If you don't run `/etc/catman`, SPP-UX formats each man page the first time a user calls it up via the `man` command. The formatted version is added to the appropriate `cat` directory and used in subsequent accesses.

   If you decide to use this method, you must make directories to hold the formatted manpages. The following script creates these directories:

```
cd /usr/man
for num in 1 1m 2 3 4 5 7 8 9
do
 mkdir cat$num
done
```

   When all the manpages have been formatted, you can remove the `nroff` source files.

Adding users and groups

If you are setting up your system for the first time, it is now time for you to add *users* and *groups* to the system. These are the names of the data structures by which an SPP-UX system recognizes a given person or class of people who use it.

There are SAM screens for setting up users and groups, or you can do it by means of SPP-UX commands. Complete directions are in Chapter 5.

Setting up electronic mail

Electronic mail (or *email* as it is often called) can be run by any of these three utilities: `elm`, `mailx`, or `mail`. Any user can use any one of these utilities.

- If your users will be exchanging messages only with each other, and will not need to send mail

to users on other systems in a network, then you need not do any setup.

The mailer will do the initialization needed for each user when the user first invokes the mailer. However, you may want to supply each `mailx` and `elm` user with a customization file, setting up useful defaults. Depending on the mailer, the customization file should be:

`mail:`	(none)
`mailx:`	`$HOME/.mailrc` (that is, a file named `.mailrc` in the user's home directory) (In addition, `mailx` uses a system-wide `/usr/lib/mailx/mailx.rc` defaults file)
`elm:`	`$HOME/.elm/elmrc`

- If your users will be sending and receiving mail over a network, you need to set up routing through ARPA Services. To configure ARPA Services, follow the directions in the manual *Installing and Administering ARPA Services*. You will also need to install the ARPA Services `sendmail` utility. Chapter 6 of the *Installing and Administering ARPA Services* manual contains the directions you'll need.

Setting up the line-printer spooler

You share printers among users via the line-printer spooler, which intercepts print requests, organizes them into a queue, and feeds them to the printer one by one. A printer that has been configured into the line-printer spooler is referred to as a *spooled* printer. Any printer SPP-UX supports can be spooled.

If your system will have more than one user at any one time, you should spool the printer(s); if you don't, any listing sent to the printer while another listing is printing will be interleaved with it, garbling both listings.

If your system is part of a network, the line-printer spooler also lets you send print requests to, or receive them from, other computers in the network,

allowing you to make the most efficient use of your printers.

Setting up the line-printer spooler is one of the more complicated tasks you need to do at this stage. Chapter 7 contains full explanations and directions. If you have not administered an SPP-UX system before, or have not set up a spooled printer before, read the chapter carefully before you attempt the task.

Setting up news

news is a utility that allows you to post messages for users to read.

- To create a news item, create a file with your text editor and place it in the directory /usr/news.
- To make sure users know about news items they haven't read yet, do the following:

  For Korn and Bourne shell users, edit /etc/profile to include the following statement:

```
if [ -f /usr/bin/news ]
then news -n #notify if news.
fi
```

  For C shell users, edit /etc/csh.login to include the following:

```
if ( -f /usr/bin/news ) then
   news -n #notify if news.
endif
```

When they log in, if there are news items they haven't read, users will see a message like this:

```
news: news_filename
```

where *news_filename* is the name you gave the file in /user/news.

Users can enter news and the item or items will print on the screen. For more information, see the news(1) man page.

Setting up system accounting

System accounting allows you to:

- Monitor individual users' disk space usage
- Record logins and logouts
- Collect data about individual processes, such as memory usage and execution time
- Charge fees for usage
- Generate summaries and reports that you can use to analyze system performance and bill users for resource consumption

If you need to set up system accounting, you should do so now. Details are in Chapter 8.

Customizing the system

Customizing the system usually means editing a file, either to change the way the system behaves in general, or to modify the way a particular user interacts with it.

The most important files you can customize are:

`/etc/inittab` Contains information about system run levels and also has an entry for each terminal.

`/etc/rc` Defines actions taken during system startup.

`/etc/passwd` Determines who can log into your system.

You can add, delete and modify entries either by editing the file, or by means of SAM screens under the Users menu. Chapter 4 contains more information.

`/etc/group` Identifies the users that form a group, associates group IDs (GIDs) with group names, lists users, and associates those users with a group name and a group ID. There's more information in Chapter 4.

`/etc/ttytype` Used by the `tset` command as a database of terminal types on your system.

Edit this file when you add a new type of terminal or modem to your SPP-UX system. For example:

```
300h console
2397 tty00
2397 tty01
```

`.exrc` Maps terminal characteristics and sets up key definitions for the ex family of SPP-UX editors (`vi`, ex, and so on).

`/etc/issue` Determines what the user will see before the login prompt.

`/etc/motd` Contains the message of the day.

`/etc/profile, /etc/csh.login` Executes automatically during the login process.

The `/etc/profile` file executes for Bourne, Korn, and restricted shell users. The `/etc/csh.login` file executes for C shell users.

`$HOME` files (Files in the user's home directory).

`.profile` Executes each time the user successfully logs in using the Bourne shell, Korn shell, or restricted shell.

`.kshrc` Korn shell script that supplements actions taken by the `.profile` file.

Executes whenever a new Korn shell is spawned, if specified by the following statements in the user's `.profile`:

```
ENV=$HOME/.kshrc
export ENV
```

The name **.kshrc** is merely a convention: whatever file you specify will execute.

`.cshrc` executes when a new C shell starts.

`.login` executes when a C shell user logs in, after `.cshrc`.

Customizing system startup

When the system boots, it executes a series of programs and shell scripts. (Details are in Chapter 4.) Of the files involved, you can customize `/etc/rc` and `/etc/inittab`.

Editing the /etc/inittab File

`/etc/inittab` is input to `/etc/init`, the first program SPP-UX runs after obtaining control from the boot ROM. Use `/etc/inittab` to set system run-levels.

You need to edit `/etc/inittab` whenever you add a new terminal to your system. An entry for a terminal whose device file name is `/dev/tty04` would look like this:

```
04:2:respawn:/etc/getty tty04 H
#comment to identify user
```

When you start up the system, this terminal will receive a `login:` prompt, and the prompt will be redisplayed after the user logs out.

Editing the /etc/rc File

The `/etc/rc` script is executed by the `/etc/init` program during system startup.

`/etc/rc` performs a number of functions, including setting the timezone and the date, and initializing system processes such as the syncer daemon and the line-printer spooler.

Edit `/etc/rc` to do any processing you might want this particular system to do when it boots.

For example, if you have Network Services, you might want to start the proxy server here.

You should put this processing in a separate script, such as `/etc/rc.local`, and call the script from `/etc/rc`. This way, it is easier to recreate your customization if `/etc/rc` is overwritten when you update SPP-UX to a new release.

Customizing the login prompt (etc/issue)

`/etc/issue` contains textthat users will see immediately before the `login:` prompt. Normally it identifies the system (by the host name from `/etc/hosts` if this is a networked system, and its friendly alias, if any), the release of SPP-UX, and includes any other information you want to put there.

For example:

```
Parsec [SPP-UX Release 2.1 SPP-1000XA]
```

Posting a message of the day (/etc/motd)

The message of the day appears each time a user logs in if the user's personal customization file (`/etc/profile` for Bourne and Korn shell users or `/etc/csh.login` file for C shell users) has the following line:

```
cat /etc/motd # message of the day
```

Edit `/etc/motd` to display topical messages. For example:

```
Scheduled power outage in Bldg. 801
8AM-noon Saturday; power off all
equipment in Bldg. 801 when you
leave on Friday.
```

Customizing users' login environments

`/etc/profile` and the `.profile` file in the user's home directory execute when a Bourne or Korn shell user logs in. When a C shell user logs in, `/etc/csh.login` executes, and so do the `.cshrc` and `.login` files in the user's home directory.

`/etc/profile` and `/etc/csh.login` should contain the defaults for variables such as the timezone setting, the terminal type, search path, and mail and news notification. These can be overridden if necessary in individual users' `.profile` or `.login` files.

The `.cshrc` file in the user's home directory performs additional set-up such as setting aliases (user-defined commands). `.kshrc` performs similar tasks for a Korn shell user if it is declared in the ENV variable.

When you add a new user, you may want to place default versions of these files in the user's home group. (If you use SAM to add a user, SAM puts the appropriate files in the home group for you.) You can use the sample files in the `/etc` directory (such

as /etc/d.profile) as templates, editing them as you wish.

Customizing users' editing environments

This means editing the .exrc file in the user's home group to enable keyboard features such as the cursor arrow keys, and to set other options in the ex family of editors, including vi.

The .exrc file functions only if the EXINIT environment variable is not defined in the /etc/profile or $HOME/.profile files.

/etc/d.exrc is a sample .exrc file. You may want to customize the file and provide it to new users as a default.

Setting up non-standard terminal types

Files in directories under /usr/lib/terminfo enable your users to use a wide variety of terminals.

To set a user up with a non-standard terminal, do the following:

Step 1 Make sure the fileset NONHPTERM has been loaded:

```
ls /etc/filesets/NONHPTERM
```

If the fileset is not there, you can get it from your latest SPP-UX release media (the media on which you got the current release of SPP-UX).

Run the swinstall program and select the NOHPTERM fileset. The *Exemplar Software Installation* manual shows how to load an individual fileset.

Step 2 Find the file that corresponds to the terminal you want to set up, if the file exists.

Suppose you want to set someone up with a Wyse (TM) 100 terminal. All supported terminals whose names begin with "w" are listed under /usr/lib/terminfo/w.

Enter

```
ls /usr/lib/terminfo/w
```

and you'll see an entry called `wy100`. This is the terminfo file for the Wyse 100.

If there is no `terminfo` file for the terminal you want to add, you can create one. See the section "Creating a new terminfo file" on page 24 for instructions.

Step 3 Find the terminal name in the file.

For example,

```
more /usr/lib/terminfo/w/wy100
```

This will produce a screenful of special characters, but near the beginning you'll see `wy100|100|wyse 100`. This means you can refer to the Wyse 100 by any of the names `wy100`, `100` or `wyse 100`.

Step 4 Set the user's TERM environment variable in the appropriate login script in their home directory: `.profile` for a Korn or Bourne shell user, or `.login` for a C shell user.

For example (Bourne or Korn shell):

```
TERM=wy100
export TERM
```

(C shell):

```
set TERM wy100
```

The default versions of these scripts prompt the user for the terminal type when he or she logs in, so rather than editing the script, you could simply tell the user to respond with the terminal name, for example:

```
TERM = (hp) wy100
```

Creating a new terminfo file

If there is no terminfo file for the terminal you want to set up, you can create one. The `terminfo(4)` man page explains the rules for constructing a `terminfo` file.

You may want to copy an existing `terminfo` file. In this case, get into the directory containing the file you want to copy and create an ASCII version of the file.

For example, to make a copy the file `/usr/lib/terminfo/w/wy100`, do the following:

Step 1 Log in as superuser.

Step 2 Change directories:

```
/usr/lib/terminfo/w
```

Step 3 Make an ASCII version of the file:

```
untic wy100 > filename
```

where *filename* is whatever you want to call the new file. Make it similar to the terminal's product name and model if you can.

Step 4 Edit the file to reflect the capabilities of the new terminal.

Make sure you change the name(s) of the terminal in the first line. See `terminfo(4)` for rules for entries.

Step 5 Compile the new terminfo file:

```
tic filename
```

For more information on using the `terminfo` compiler, see the `tic(1M)` man page.

The Subcomplex Manager utility

3

The processors and memory in a Convex Exemplar system provide a powerful array of computing resources. The Subcomplex Manager utility allows system administrators to configure processor and memory resources to best meet the needs of system users. In addition, the Subcomplex Manager allows all users to view the current allocation of system resources.

The primary unit for allocating computing resources to Exemplar users is the *subcomplex*. A *subcomplex* is a collection of processors and memory from one or more nodes of the system. Every system user is authorized to use one or more subcomplexes. Each processor and block of global memory can belong to only one subcomplex. Every process runs within a subcomplex; a process can use only the processors and global memory allocated to that subcomplex.

Additional resources that are configurable with the Subcomplex Manager are each node's *server set*, each node's *CTI cache*, and the *system subcomplex*.

Each node has a set of one or more processors called the *server set*. The processors in the server set handle all operating system activity for the node. A processor can be in the server set and in a subcomplex simultaneously. Processors in the server set handle system activity from all processors on the node without regard to subcomplex boundaries. By default, all processors on a node are assigned to the server set. When you create a subcomplex, you can choose whether the processors in the subcomplex remain in the server set.

Each node has a portion of its global memory allocated to the node's coherent toroidal interconnect *(CTI) cache*. The CTI cache contains

coherent memory data fetched from other nodes on the system (there is no reason to establish a CTI cache on a single-node system). The default value for the CTI cache size is specified in the Open Boot parameters; see the *Exemplar Open Boot Quick Reference* for information about setting this value. You can specify an amount of global memory to be used for each node's CTI cache. There is no partitioning of CTI cache on a node; the CTI cache is shared by all processes on the node, and any process may use as much of the CTI cache as it needs.

The *system subcomplex* is a special subcomplex that is created automatically at boot time to run system processes, including `init` and processes spawned by `init`. System administrators can modify the system subcomplex, but must leave at least one processor on the root node (node 0) allocated to the system subcomplex.

Subcomplex Manager interfaces

The Subcomplex Manager utility provides a both a graphical user interface (GUI) and a command interface. The command interface is useful for loading a system configuration when the system is booted or when the system is reconfigured automatically, such as at scheduled times of the day. The graphical user interface provides a convenient way to view system resources and to create complex configuration files.

The Subcomplex Manager graphical user interface

The Subcomplex Manager graphical user interface allows you to perfom the following functions:

- View information about system processor and memory resources and the current organization of subcomplexes
- Create and save a complex configuration file
- Load or overlay a complex configuration file into the Subcomplex Manager
- Apply a complex configuration to the system (system administrators only)

Viewing information about system resources

The Subcomplex Manager main window displays general information about system subcomplexes and processors. This window lists the subcomplexes that are running on the system, and shows which processors are allocated to each subcomplex. If you have modified any of the information in the main window, you can revert to the current complex configuration display by selecting the **Revert** option from the **File** menu.

To view detailed information about a particular subcomplex, left-click on the name of that subcomplex in the subcomplex list on the left side of the main window. This causes the Subcomplex Attributes dialog box to appear.

To view detailed information about a particular node, left-click on the name of that node in the node display on the right side of the main window. This causes the Node Attributes dialog box to appear.

You can use the **View** menu to change the focus of information in the main window from subcomplexes to node server sets.

Creating and saving a complex configuration file

A complex configuration file contains a set of specifications for allocating processors and memory to subcomplexes, as well as specifications for node CTI cache and node server sets. You can create a and save a complex configuration file by changing the settings in the Subcomplex Manager interface and saving these settings to a file by selecting the **Save ...** option or the **Save As ...** option from the **File** menu. For more information about changing the settings, see the descriptions of the Subcomplex Manager windows.

Loading a complex configuration file into the Subcomplex Manager

You can load a previously saved complex configuration file into the Subcomplex Manager and then modify that configuration or apply it to the system. To load a complex configuration file, select the **Load ...** option from the **File** menu, then specify the name of the complex configuration file you wish to load.

Overlaying a complex configuration file into the Subcomplex Manager

You can overlay a previously saved complex configuration file onto the contents of the Subcomplex Manager main window and then modify that configuration or apply it to the system. To overlay a complex configuration file, select the **Overlay ...** option from the **File** menu, then specify the name of the complex configuration file you wish to overlay. The contents of the configuration file are merged with contents of the Subcomplex Manager main window. This operation only succeeds if no processor is specified as allocated in both the configuration file and the window.

Performing a system reconfiguration

System administrators can apply a new complex configuration to the system. The configuration currently displayed in the Subcomplex Manager is applied to the system. To perform this function, select the **Perform Reconfiguration** option from the **File** menu.

The Subcomplex Manager returns the following information when you select the **Perform Reconfiguration** option:

- A summary of all the changes that will be made to the system.
- A list of subcomplexes that must be idled before the new configuration can be applied. This list will only appear if you attempt to remove all the processors for a subcomplex from a node; in this

case, you must kill all processes running in that subcomplex.

- If the configuration cannot be applied to the system, the reasons will be listed. For example, if you attempt to remove all processors from the system subcomplex, the Subcomplex Manager will inform you that the configuration cannot be applied to the system for that reason. For a list of reconfiguration error messages, see the *Error Messages* section.

If the configuration can be applied to the system, you are given the opportunity to kill the necessary processes, then perform the reconfiguration operation.

Subcomplex Manager windows

The Subcomplex Manager graphical user interface consists of the following windows:

Subcomplex Manager main window

This window appears when you run SCM. It displays a list of subcomplexes on the left side, and a picture of all the nodes and processors in the system on the right side. This window displays the current complex configuration when you start the interface (unless you specified the `-n` option in the `cmgr` command); you can then use this window as a work area to create a modified complex configuration.

The Subcomplex Manager main window consists of the following elements:

Subcomplex list

The subcomplex list, which appears on the left side of the window by default, displays the subcomplexes that are used in the current configuration, along with an icon representing each subcomplex. You can add and delete subcomplexes from this list. You can perform the following actions on this list:

Select a subcomplex name

This action activates the Subcomplex Attributes dialog box, which allows

you to view and edit detailed information about the subcomplex.

Select a subcomplex icon

This action assigns all processors that are currently selected in the Node display to the subcomplex you have selected.

Node display

The node display on the right side of the window shows the processors on each node of the system, and the subcomplex assignment of each processor. Processors are displayed in rows of two; the processors in each row share a single CPU agent board. You can perform the following actions on this display:

Select a node name

This action activates the Node Attributes dialog box, which allows you to view and edit detailed information about the node.

Select processors

This action highlights a set of processors and allows you to assign them to a subcomplex (by selecting the subcomplex's icon from the Subcomplex list), or to assign them to or remove them from their node's server set (if the Server set list is currently displayed in the left side of the window). You can select a single processor by left-clicking on it; you can select a contiguous group of processors by *rubber-banding* (outlining a rectangle that includes all of the processors while holding down your left mouse button). Normally, selecting a processor or group of processors deselects all other processors; if you hold down your Shift key while selecting a processor or

group of processors, any processors already selected will remain selected.

Menu bar

The menu bar along the top of the window allows you to perform a variety of functions on the contents of the Subcomplex Manager main window. The menu bar contains the following pull-down menus:

File

New ...	Clears all subcomplexes from the window, allowing you to create a new machine configuration from scratch
Load ...	Prompts you for the name of a configuration file, and replaces the contents of the window with the contents of the configuration file
Overlay ...	Prompts you for the name of a configuration file, and merges the contents of the window with the contents of the configuration file. This operation only succeeds if no processor is specified as allocated in both the configuration file and the window.
Save ...	Saves the configuration currently displayed to the current file name.
Save As ...	Prompts you for a file name, and sves the configuration currently displayed to that file
Perform reconfiguration	Reconfigure the machine using the configuration displayed in the window
Revert	Replaces the contents of the window with the current machine configuration
Quit	Exits the Subcomplex manager

Edit menu

`Select All` Selects all processors on all nodes in the Node display portion of the window.

Subcomplex menu

`Create` Activates the Subcomplex Attributes dialog box containing information for a new subcomplex; you can edit this information and then select `Ok` to create a new subcomplex

`Remove` Activates a popup dialog box that allows you to select a subcomplex to remove from the Subcomplex list and the Node display

Node Attributes dialog box

Displays the resource allocations for a selected node. For each subcomplex to which processor on the node have been assigned, the amount of global memory and the requested amount of CTI cache on the node are shown. You can perform the following actions from this dialog box:

Allocate global memory

Enter the amount of the node's global memory, in MBytes, to be allocated to this subcomplex.

Allocate CTI cache

Enter the amount of CTI (coherent toroidal interconnect) cache, in MBytes, for the node. You must select one of the values listed in the menu that pops up when you click on the CTI cache button; the values that can be selected are 0, 16, 32, 64, 128, 256, 512, and 1024 MBytes.

You must make sure that the sum of the global memory and CTI cache allocations does not exceed the amount of global memory on the node; an attempt to reconfigure the machine will fail if a node's global memory is overallocated.

When you have finished changing the values, you can select the `Ok` button to make the changes or select the `Cancel` button to cancel the changes.

Subcomplex Attributes dialog box

Displays information about a single subcomplex, including the subcomplex name, ownership and permissions, scheduling policy, default memory allocations, and pixmap icon. You can change the following subcomplex attributes:

`Subcomplex Name`

A unique name identifying the subcomplex. Subcomplex names are limited to 32 characters.

`User` The owner of the subcomplex. This value must be a valid user login ID. This value is used in conjunction with permission information to control access to the subcomplex. Use the user ID `root` in this field if you wish to limit ownership to system administrators.

`Group` The user group that can use the subcomplex. This value must be a valid group ID to which the subcomplex owner belongs. Will be used in conjunction with permission information to control access to the subcomplex; generally, this is the group of users who will be allowed to run processes on the subcomplex (indicated by execute permission).

Permissions `Read`, **`Write`**, and **`Execute`** permissions for the user, group, and world (other). Read permission indicates that the user or group is allowed to view the subcomplex definition. Write permission indicates that the user or group is allowed to modify the set of enabled scheduling policies for the subcomplex. Execute permission indicates that the user or

group is allowed to run processes on the subcomplex.

Default Global Memory

The amount of the node's global memory, in MBytes, to be allocated to this subcomplex.

`Icon` Select the icon shown to view a list of icons from which to choose an icon to represent this subcomplex. The default icon is a black square with a numeral *n* in the lower left corner indicating that this is the *n*th subcomplex defined in the current machine configuration.

`Fixed Priority Scheduling`

Select this field to enable fixed friority scheduling.

Reconfiguration constraints

There are some constraints which apply when reconfiguring the complex while it is in use, or when making changes to a subcomplex which is currently loaded on the complex (even if it is not yet in use). A subcomplex which has one or more user processes currently assigned to it is "busy".

The constraints are as follows:

- A busy subcomplex cannot be removed from the complex without first killing all user processes on that subcomplex. The Subcomplex Manager will not kill processes; the user must do this explicitly using the 'kill' utility.
- The global memory of a subcomplex cannot be reconfigured once any has been added
- If a subcomplex already has global memory allocated, then the user cannot add new processors unless they belong to one of the nodes which are already being used by the subcomplex. This constraint applies even if the subcomplex is not currently busy. This results from the implementation of "global memory domains"; for further information, see the Physical Memory Design Specification.

- The system subcomplex can never be removed from the complex, and it must have at least one processor on the "root node" (node 0) assigned to it at all times. This subcomplex is created automatically on the root node when the complex is booted.

The following reconfiguration actions *are* allowed on a busy subcomplex:

- Processors may be added to or removed from a subcomplex at any time (except as noted above, if global memory has been allocated).
- Removing all processes for a particular subcomplex from a node will suspend all threads running in that subcomplex on the node.
- Subcomplex attributes including name, uid, gid, and permissions may be changed at any time.
- Scheduling policies may be enabled or disabled on a subcomplex at any time.
- Processors may be added to or removed from the server set on a node at any time, provided that at least one processor remains in the server set at all times on each node.

Global memory regions are contiguous in physical memory. It is not possible to create global memory for a subcomplex on a node if a sufficiently large contiguous region of free memory does not exist on the node. Creation of a large global memory region is most likely to succeed immediately after booting the system.

Error messages

The Subcomplex Manager issues the following error messages:

`create` *subcomplexid*

A subcomplex with subcomplex ID *subcomplexid* has been created.

allocate node nodeid cpu cpuid scid subcomplexid

CPU cpuid on node nodeid has been allocated to subcomplex subcomplexid.

deallocate node nodeid cpu cpuid

CPU cpuid on node nodeid has been removed from the subcomplex to which it was previously assigned.

server set ssign node nodeid cpu cpuid

CPU cpuid on node nodeid has been assigned to the node's server set.

server set remove node nodeid cpu cpuid

CPU cpuid on node nodeid has been removed from the node's server set.

The Subcomplex Manager command-line interface

The `scm` command-line interface allows you to manage subcomplexes in a non-interactive mode. The `scm` command has the following syntax:

```
scm [ -n nnodes -c | -l filename |
-o filename | -r scname | -sc]
```

If no options are specified, or if only the `-n` option is specified, the graphical user interface is activated. If any of the other options is specified, the command is run without the GUI.

scm command options

The following command line options are recognized by `scm`:

`-c`	Return a full description of the current system configuration. If you are not running with superuser privileges, only the information about subcomplexes for which you have read access is returned.
`-l` *filename*	Load the complex configuration from *filename* to the system Requires superuser privileges
`- n` *nnodes*	Perform all operations on a hypothetical machine with *nnodes* nodes. This option disables the `-l`, `-o`, and `-r` options and their equivalent functions in the GUI.
`- o` *filename*	Overlay the complex configuration from *filename* to the system. Requires superuser privileges.
`-r` *scname*	Remove subcomplex *scname* from to the system. Requires superuser privilege.s
`-sc`	Return a list of the names of all subcomplexes currently loaded on the system. Only those complexes for which the user has read access are displayed.

scm command examples

To see a description of the current machine configuration, log in as the superuser and enter the following command:

```
scm -c
```

To load the complex configuration file `weekendconfig` onto the system, log in as the superuser and enter the following command:

```
scm -l weekendconfig
```

To create a machine configuration file for a hypothetical system containing 16 nodes (128 processors) using the `scm` GUI, enter the following command:

```
scm -n 16
```

Subcomplex manager configuration file format

The SCM congifuration file is created automatically by the SCM GUI. This file can be stored anywhere in the file sys tem. The file can also be created or edited manually.

The configuration file consists of a list of subcomplex definitions. All of the text following a subcomplex SC keyword, up to the next SC keyword, describes that subcomplex.

All statements in the configuration file have the form:

```
KEYWORD value
```

Except for the `SC` and `PIXMAP` keywords, *value* must be an integer value.

The SCM configuration file can contain the following keywords:

SC
: The name of the subcomplex. All statements following this statemsnt, up to the next SC statement, are part of this subcomplexs definition. The value must be a character string from 1 to 32 characters long.

UID
: The UID of the owner of the subcomplex. This value, along with the subcomplex permissions, determines the actions the owner can perform on the subcomplex. Generally, the owner is the only user with write permission to the subcomplex. The default is the UID of the user who invokes the `scm` command.

GID
: The group ID of the group that uses the subcomplex. This value, along with the subcomplex permissions, determines the actions the group can perform on the subcomplex. Generally, this group has execute permission (permission to execute processes on the subcomplex). The default is the effective group ID of the user who invokes the `scm` command.

PERM
: The read, write, and execute permissions for the subcomplex owner, user group, and the world. The value must be a four-digit octal number; the first digit should be 0. The remaining digits represent the

	permissions for the owner, group, and world, respectively; the values are the same as those for the chmod1 command. The default value is 0754 (read, write, and execute permission for the owner, read and execute permission for the group, and read permission for the world).
POLICY	This value indicates whether fixed-priority scheduling or timeshare scheduling is enabled for this subcomplex. The values for this keyword are defined in `/usr/include/sys/cnx_tattr.h`.
PIXMAP	The name of the file containing the pixmap for the icon to be used in the SCM GUI for this sub-complex. The default is to let SCM generate a unique icon for the subcomplex.
NODEID	The physical node ID of one of the nodes that has processors allocated to this subcomplex. The value can range from 0 through one less than the number of nodes in the system.
PROCID	The physical processor ID for one of the processors on the node specified in the previous NODEID entry. The value can range from 0 through 7.
SERVERSET	This value indicates whether the processor specified in the previous PROCID entry is in the nodes server set. A value of 0 indicates that the processor is not in the server set; a value of 1 indicates that the processor is in the server set. The default value is 1.
GMEM	The amount, in megabytes, of global memory to be allocated to this subcomplex on the node speci- fied in the previous NODEID entry. The default value is 0.
CTICACHE	The amount, in megabytes, of CTI (coherent toroidal interconnent) cache to request for this subcomplex on the node specified in the previous NODEID entry. The value must not exceed the amount of global memory present on the node. The default value is 0.
DEFAULT_GM	The default amount, in megabytes, of global memory that will be allocated to processors on the node specified in the previous NODEID entry if they are added in the future. The default value is 0.

Example SCM configuration file

The following is a definition of a subcomplex named zymurgy. This subcomplex contains two processors each; processors 6 and 7 from nodes 0 and 1. 128 MB of global memory is allocated to this subcomplex; both nodes have 16 MB of CTI cache. The two processors on node 0 remain in their node's server set; the two processors on node 1 are not in their node's server set. The subcomplex is owned by the user with UID 26645, and can be used by the user group with GID 524.

```
NODEID=0: CTICACHE=16:
NODEID=1: CTICACHE=16:
SC=zymurgy:
    UID=26645: GID=524:
    NODEID=0:
        GMEM=128:
        PROCID=6:
        PROCID=7:
    NODEID=1:
        GMEM=128:
        PROCID=6:SERVERSET=0:
        PROCID=7:SERVERSET=0:
```

Starting and stopping SPP-UX

4

Starting and stopping SPP-UX are routine tasks, but they are critical to the operation of your computer. When the system is turned on, you can allow the default operating system to boot, or select other boot options. When you stop a system, you must use the appropriate shutdown process. Simply turning the system off with the power switch can corrupt the file system. When you change the system to an administrative (single-user) state, during shutdown, you can reboot (restart) the system without turning it off, or you can shut the system down completely.

This chapter describes:

- Turning on the Exemplar system
- Starting (Booting) SPP-UX
- Setting initial information
- Stopping the system

Before starting your Exemplar system for the first time, make sure that you have performed all the instructions provided in the *Exemplar Site Preparation* manual and the *Exemplar Installation Guide.*

The following sections describe how to start SPP-UX if you are starting with a system that has everything connected but not turned on.

Turning on the system

Turn on the Exemplar test station, set up a working environment on the test station, and perform all of the test station installation instructions provided in the *SPP-UX Installation and Upgrade* manual before turning on the Exemplar system.

When you turn on the Exemplar system, system diagnostics are run before SPP-UX is booted. When the SPP-UX microkernel boots, a System Console window will appear on the Exemplar test station. You can begin to control SPP-UX at this point.

During the boot sequence, use the ESCAPE key to terminate or interrupt the current process. The Open Boot software controls the boot process until the entire operating system is finished loading.

If you want to run the Exemplar system in single-user mode, press the ESCAPE key, then enter the following command in the System Console window:

```
boot -s
```

If you do nothing to interrupt the boot process, the SPP-UX operating system automatically boots to multiuser mode. See the *Exemplar Open Boot Quick Reference* for other operations you can perform while the system is booting, and for information on how to configure the boot process on your system.

Watch the startup messages. Compare what starts up with what you expect, and note possible problems. The exact messages depend on your configuration. The startup process ends when you see the login prompt. If you do not get the prompt, the system did not start up. You will need to determine why. During the startup process, the system will perform a file system consistency check of the root disk if the system was shut down improperly.

Starting SPP-UX

You must start up (boot) SPP-UX when the operating system has been completely shut down (as is required before you turn the computer off) or after you have partially shut down the operating system to perform system administration tasks.

Here are some guidelines:

- You must reset the system for it to boot. Reset the system by turning on the main power. The operating system will start to boot, and if this is the first time that the system has been booted,

you will be prompted for initial system parameters. See "Setting system parameters" on page 11 for details.

- Some SAM tasks and SPP-UX commands may restart (reboot) the system for you (for example, `/etc/reboot`).
- To start your SPP-UX system, you must have configured and installed the hardware and the software. See other chapters in this manual, as well as the *Exemplar Installation Guide* and the *SPP-UX Installation and Upgrade* manual for more information.
- Your system must have certain files to start up properly (for example, `/etc/init`, `/etc/inittab`, `/dev/console`, `/etc/bcheckrc`, `/etc/brc`, and `/etc/rc`). Without these files, the startup process will fail. Open Boot hands off control to the `/etc/init` process, which sequentially executes the contents of `/etc/inittab`. `/etc/bchecrc`, `/etc/brc`, and `/etc/rc` are scripts specified in the `/etc/inittab` that must be executed to check the file system and initialize your system. See the "Reviewing the State of the File System" section of this chapter for details. Errors found in the file system might require you to perform additional tasks.
- The operating system will continue to boot up and will display additional information about your system. After the bootup process has completed, a login prompt will appear in the System Console window.

You are now ready to use your SPP-UX system.

Reviewing the state of the file system

During the startup process, the `/etc/bchecrc` script executes `/etc/fsclean`. This command determines the shutdown status of the system and returns three possibilities:

1. If the file systems were shut down properly, the startup process continues and you see the following message:

```
/etc/fsclean:/dev/dsk/0s0(root
device) ok file system is OK,
not running fsck
```

2. If any file systems were not shut down properly, the startup process is interrupted and you see:

```
/etc/fsclean:/dev/dsk/0s0 not
ok run fsck FILE SYSTEM(S) NOT
PROPERLY SHUTDOWN, BEGINNING
FILE SYSTEM REPAIR.
```

   At this point, the system runs `/etc/fsck` in a mode that can correct certain inconsistencies in the file systems without your intervention and without removing data. The `fsck` command will either:

   - repair and reboot the system, incorporating the changes, or
   - you may be asked to run the `fsck` command manually. If you need to run `fsck` manually, see Chapter 6.

3. If `fsclean` detects any other errors (for example, not being able to open a specified device file), you get an error message. The startup process can end, and you will need to solve the problem.

Shutting down the system

Typically, you shut down the system down to:

1. Put it in single-user state so you can update the system or check the file systems.
2. Turn it off totally so you can perform a task such as installing a new disk drive.

Caution

Stopping the system improperly can corrupt (damage) the file systems. Never stop the system by turning it off.

An overview of shutdown

- Only the system administrator or a designated superuser should shut down the system.
- You can use SAM to shut down the system.
- The /etc/shutdown command warns users of impending shutdown, halts daemons, kills unauthorized processes, unmounts file systems, puts the system in single user mode, and writes the contents of the I/O buffers to a disk.
- The shutdown command warns all users to log off the system, using a grace period you can specify.
- Some system administration tasks will do the rebooting for you.
- If you use a network service, do not run `shutdown` from a remote system via `rlogin`. The shutdown process logs you out prematurely and returns control to the system console. Run `shutdown` from the System Console window on the Exemplar test station.
- See the `shutdown(1M)` man page for information about options and features.

Going to the single-user mode for maintenance

You must be a the system administrator with superuser capabilities or a designated user with superuser capabilities to shutdown the system.

Step 1 Change to the root directory.

```
cd /
```

Step 2 Shut down the system.

The `shutdown` command warns all users to log off the system, using a grace period you can specify.

If you do not specify a grace period, users get 60 seconds to log off. You should notify active users as to when the system will be shut down. Give them enough time to finish their work and log off. You can

do this physically or use the `/etc/wall` or `/etc/cwall` commands.

The following examples show alternatives for shutting down to the single-user state:

`shutdown`	Shuts down to single-user state, allowing the default 60-second grace period
`shutdown 0`	Shuts down the system with no grace period
`shutdown 30`	Begins the shutdown to the single-user state after a 30-second grace period

As always, watch the messages to see that everything is happening correctly.

Step 3 While the system is in the single-user state, perform the necessary system administration tasks. When you finish, you can start up the system without turning off anything by executing:

```
/etc/reboot
```

Shutting down the system completely

You must be a the system administrator with superuser capabilities or a designated user with superuser capabilities to shutdown the system.

Stopping the system improperly can corrupt (damage) the file systems. Never stop the system by turning it off.

Step 1 Change to the root directory.

```
cd /
```

Step 2 Shut down the system. It is best to shut down the system in two steps:

1. Execute:

shutdown 20	This gets the system into the single-user state, allowing a 20-second grace period

   The shutdown process asks if you want to send a message. If you elect to broadcast a message, respond with <computer | y | and then type the

message. When you finish, press [[Return]] (or [[Enter]]), and then [[CTRL]]-[[d]].

2. Execute:

   `reboot -h` This brings the system to a complete halt

   You see several messages during the process. You should watch them to note actions and possible problems. You know the system is shut down completely when the system displays `halted` and pressing a key in the System Console window takes no action.

Step 3 When the system is halted, turn the system off as follows:

1. If you have only a computer (no expander), turn the computer off. Then, turn the devices off as required.

2. If you have a computer and an expander, turn the computer off, turn the expander off, and then turn the devices off as required.

Step 4 When you want to start up the system again, see the earlier procedure for starting up SPP-UX.

Note

From the multi-user state, you can also shut down the system completely by executing:

```
shutdown -h
```

This process is more harsh and sudden than the two-step procedure described above.

Designating system shutdown authorization

You can designate which users are authorized to run `shutdown` by listing these users in the file `/etc/shutdown.allow`. If this file is empty, only the superuser has shutdown authority, but if this file is not empty, and the superuser login (usually `root`) is not listed in the file, then the superuser will not be permitted to shutdown the system. In a non-empty `shutdown.allow` file, only those users listed will have shutdown authority.

Customizing the shutdown process

You can customize the shutdown process by placing Bourne shell scripts in the file `/etc/shutdown.d`. These scripts will be executed in an ASCII (machine-collated) order. These scripts are optional, and are not required to run `shutdown`.

Power failure considerations

A local power failure means a power failure that halts the computer by affecting its central bus.

Remote power failures (affecting a remote bus) or device power failures (affecting a device) do not affect the system as a whole, unless the remote devices provide a vital system resource.

Power failure-related tasks

If you know power is going out soon: Shut down the computer and turn off power.

If a local power fail occurs: If possible, turn off all computer equipment affected by a power failure until power is completely restored. An electrical surge as power is coming back on could seriously damage hardware that has been left turned on.

Using SAM to halt or reboot the system

You can use SAM to reboot or shut down your Exemplar system. From the SAM main window, perform the following steps:

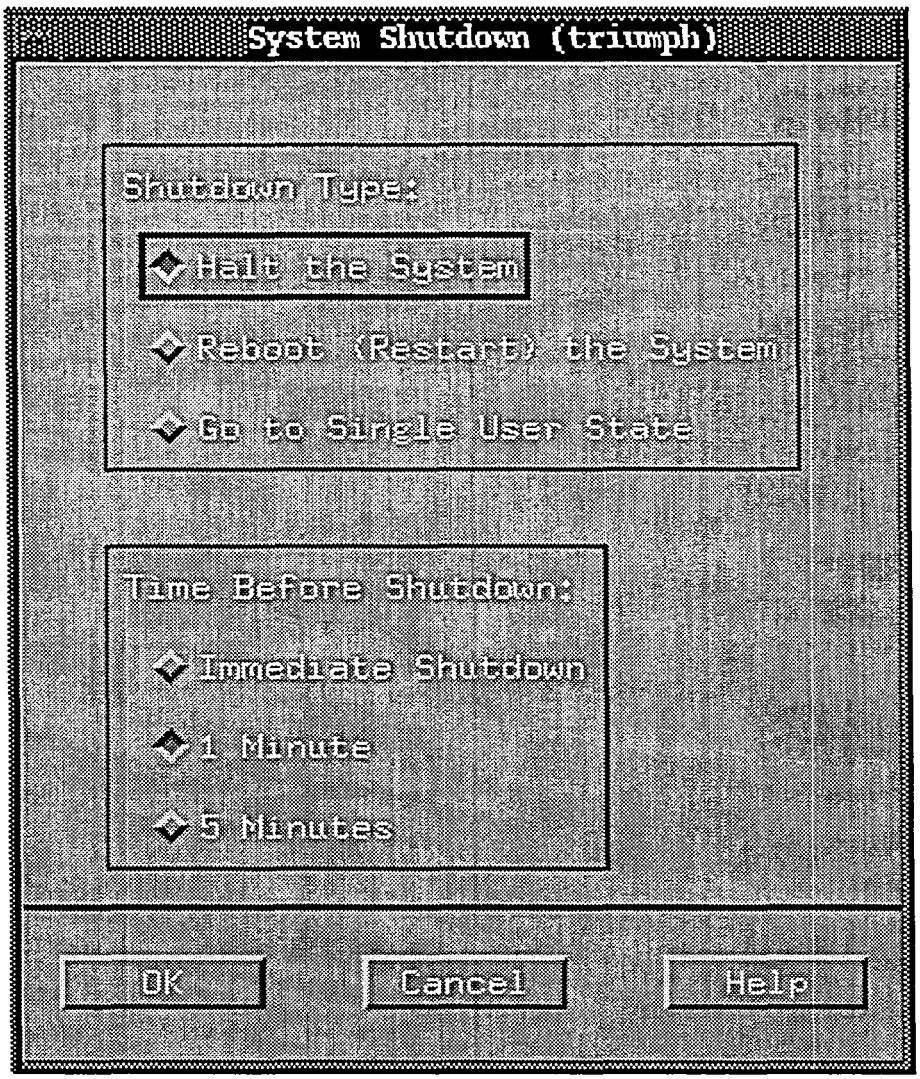
Step 1 Start SAM.

Step 2 From the SAM main window. select Routine Tasks and then select the Open button.

Step 3 From the Routine Tasks display, select System Shutdown and then select the Open button.

The window shown in Figure 4 will appear.

Figure 4 SAM System Shutdown window



Step 4 Select one of the buttons to halt (shut down) the system, reboot the system, or bring the system to single-user state. The window allows you to specify a time delay before the operation is performed.

Step 5 Press the OK button after making a selection. SAM presents you with a confirmation dialog box. If you select Yes, SAM will prompt you for a message to broadcast to all system users notifying them of the action. Enter a message, then confirm the operation one last time.

Controlling access to SPP-UX

5

It is rare to find a computer installation where everyone has access to all of the computer's files, commands and hardware resources. It is therefore likely that you will want to control who has access to your system, its data and its commands.

Authorized users gain access to the system by supplying a valid user name (login name) and password.

For additional information about the login process and the `/etc/passwd` file, set the `login(1)` and `passwd(4)` man pages.

Terms used in this chapter

access permissions — Values associated with each file that control who has permission to read, write (modify) or execute the file.

effective group — If a user changes their default or primary group with the `newgrp` command, the new current group is the effective group (see **group** and **primary group** below).

group ownership — The secondary ownership associated with each file, associating the file with a group (see **group**, below).

group — Users on an SPP-UX system can be grouped. If a group has access to a file, then any user who is defined as a member of that group will have access to the file. Users can be members of more than one group.

group_ID — Also known as GID, is a unique number associated with each group (see **group**, above) that identifies the group to SPP-UX. These group_ID numbers are defined in the `/etc/group` file.

log in Process used to gain access to the computer by supplying a user name and (if required) a password.

multi-user mode An SPP-UX mode of operation that allows multiple users to access the system simultaneously. This is the normal mode of operation for SPP-UX systems. See also single-user mode.

ownership Each file on the system has an owner. The owner controls access to the file by setting its access permissions. The owner is typically (but not always) the user who created the file.

primary group or default group A user can be a member of multiple groups, but only one of those groups is considered to be the user's primary or default group. In addition to being listed as a member of groups in the `/etc/group` file, an entry exists in the `/etc/passwd` file that indicates the user's primary group. When users first log into the system, they are affiliated with their primary group. Refer to the sections of this chapter for details.

run-level An SPP-UX mode of operation. Modes of operation are defined in the file `/etc/inittab`. The `/etc/inittab` file defines which terminals and processes are active at each run-level.

single-user mode A special SPP-UX mode of operation that restricts user input to the system console. It is usually used by the system administrator to prevent others from accessing the system during special system administration activities when it is not advisable to have other system activity (for example, when updating the operating system to a new revision).

user_ID Also known as UID, is a unique number that SPP-UX uses to identify a particular user. The user_ID number zero ("0") is used to identify the superuser. User_ID numbers between 1 and 99 are used by SPP-UX subsystems. User_ID numbers above 99 are used for normal users.

user account The environment created on the system to allow the user access. Creating a user account involves updating the system to recognize the user's login name and password. You also need to give the user access to files, system resources, and applications.

Overview of controlling access your system

Securing your data against deliberate, unauthorized access is only one reason for controlling access to your system. There are three levels of access control to your system. The following list of levels also includes reasons why you would want to control access at a particular level.

1. Controlling user accounts and groups

   By controlling who can log in to your system, you can prevent unauthorized users from running programs that consume valuable system resources (making them unavailable for the authorized users of your system). By creating and controlling groups of users, you can create unique group environments. Most systems are used for multiple purposes, and user groups allow you to customize according to multiple and varying group needs.

2. Controlling file access

   By setting the appropriate access permissions for files and directories on your system, you can prevent them from being accidentally deleted or overwritten. By setting the appropriate ownership and group ownership for files (in addition to the file permissions) on your system, you can limit their use to specific users (or groups of users).

3. Controlling run-levels

   By configuring appropriate run-levels, you can activate different groups of terminals (and processes) for different situations (such as different work shifts).

Controlling user accounts and groups

Each user is defined by an entry in the file `/etc/passwd`. The `/usr/bin/vipw` command is the recommended editor for modifying the `/etc/passwd` file. The vipw command guarantees exclusive access to the `/etc/passwd` file. The `/etc/ptmp` file is created by the `vipw`, `chsh`, `chfn`, and `passwd` commands when access to the

/etc/passwd file is granted. The /etc/vipw command requires the EDITOR environment variable to be set to vi.

If you are in single-user mode and the vipw command denies you access to the /etc/passwd file with an error message, "password file busy", delete the /etc/ptmp file and try the vipw command again. It is possible that the process that created this file terminated without removing this file. See the vipw(1M), chsh(1), chfn(1), passwd(1), and passwd(4) man pages for additional information.

Users on your system can be divided into various working groups, so that files owned by members of a given group can be shared and yet protected from access by users who are not members of the group. A user can be a member of more than one group. A group can have a maximum of 200 members.

If you prefer not to divide the users of your system into separate working groups, it is customary to set up one group (usually called "users") and assign all users of your system to that group.

Users may change their current group by using the newgrp command. The new group is referred to as the *effective group* for the user. Changing to an effective group does not alter the user's primary group entry in the /etc/passwd file. The user can return to his or her primary group by specifying no parameters or options to the newgrp command.

Group information is defined in /etc/group and /etc/logingroup, which are ASCII files that you can edit with a text editor such as vi.

/etc/group defines for each group:

- group name
- encrypted password (optional)
- numerical group identifier (group_ID)
- comma-separated list of group members by user login name

For example:

```
root:*:0:

other:*:1:

bin:*:2:

sys:*:3:

adm:*:4:

daemon:*:5:

mail:*:6:

lp:*:7:
users:*:20:john,naomil,patrickd,kersch
en,michelem,dennism,pvallis

pub:*:24:patrickd,naomil,dennism
```

A blank line in the `/etc/group` file is not allowed. If a blank line appears in the `/etc/group` file, all entries after the blank line are ignored.

`/etc/logingroup` contains the identical information, but the group name and encrypted password fields are not used. It is common practice to link the `/etc/group` and `/etc/logingroup` files together using the link command (see the `link(1M)` man page).

`/etc/group` is used by the `newgrp` command to check access privileges. If the user's login name appears in the access list of the group for which access is being requested, the access is granted thus changing the user's current group to the requested group. `/etc/logingroup` in contrast to `/etc/group` allows users listed in more than one group access to files belonging to other groups without changing their primary or effective groups.

For additional information about group related tasks, see the "Managing user accounts and group tasks" section on page 63. For more details on the `/etc/group` and `/etc/logingroup` files, see the "Adding a group using SPP-UX commands" section on page 98, and the `group(4)` man page.

Primary groups

A user can be a member of multiple groups, but only one of those groups is considered to be the user's primary or default group. In addition to being listed as a member of the group in the file `/etc/group`, an entry exists in the file `/etc/passwd`, indicating which group is the user's primary group. When users first log into the system, they are affiliated with their primary group.

To change the primary (default) group that your user is a member of, you will need to change user's entry in the `/etc/passwd` file to reflect a new *group_ID* value. The *group_ID* uniquely identifies an entry in `/etc/group` and `/etc/logingroup`. For instructions, see "Displaying/modifying a user's account information using SAM" section on page 73, and the "Changing a user's primary group using SPP-UX commands" section on page 103.

Group passwords

When a user first logs into your system, his or her default or primary group affiliation is the one pointed to by the *group_ID* entry (the fourth field in `/etc/passwd`). A user may be a member of more than one group. To change which group a user is affiliated with, a user can use the `newgrp` command. `newgrp` will require a password if the group has a password and the user does not, or if the group has a password and the user is not listed as being a member of that group (in the file `/etc/group`). If the user only needs to access files in another group, a entry in the `/etc/logingroup` would permit access to other group's files without changing the user's effective group. See the `group(4)` man page for more information.

Note

The use of group passwords is not encouraged, and they are rarely used. They encourage poor security practices.

Special groups

Commands that permit access to all of the system's resources (and files) are usually restricted for use by superusers only. Although it is possible to have more than one superuser defined for your system (see the "Adding a user using SAM" section on

page 65 and the "Adding a user using SPP-UX commands" section on page 78), you may prefer to have only a subset of the superuser's capabilities available to a group of users. There are five types of special privileges that you can assign to a group of users using the `setprivgrp` command:

RTPRIO — controls group access to the `rtprio` command and system call that allow the setting of real-time priorities.

MLOC — controls group access to the plock system call that allow processes to be locked in memory and allow the use of the `shmctl` system call SHM_LOCK parameter.

CHOWN — controls group access to the `chown` command and system call that allow changing the ownership of files on the system.

LOCKRDONLY — controls group access to the `lockf` system call that sets locks on files that are open for reading only.

SETRUGID — controls group access to the `setuid` and `setgid` system calls. The `setuid` system call changes the real user ID of a process; the `setgid` system call changes the real group ID of a process.

For additional information, see the `rtprio(1)`, `rtprio(2)`, `plock(2)`, `shmctl(2)`, `chown(1)`, `chown(2)`, `lockf(2)`, `setuid(2)`, `setgid(2)`, `setprivgrp(2)` and `setprivgrp(1M)` man pages.

Any user whose current group_ID matches the group_ID of a privileged group will have access to the special capabilities assigned to that group. A group can have any one or a combination of the special privilege capabilities. See the "Displaying/assigning special group privileges using SPP-UX commands" section on page 108 for specific instructions.

Controlling file access

All of the files on an SPP-UX system have access permissions, ownership, and group ownership associated with them. Together the permissions and ownerships determine who can access the file.

File access permissions

There are three types (modes) of file access:

read Determines who can view the file's contents. For directories, read access allows access to the directory with the cd command.

write Determines who can alter the file's contents. For directories, write access allows modify and remove privileges.

execute If the file is an executable program, the execute permissions determine who can run the program. For directories, execute access allows listing the directory contents.

There are three sources of file access:

Owner The owner is usually the person who created the file (unless ownership has since been changed using the chown command).

Group Members of the group that the file belongs to.

Other All other users on your system.

There are three commands that change file access privileges:

chmod The chmod command changes the type of access (read, write, and execute privileges) for every access source (owner, group, or other). For example, you can give the owner of the file read, write, and execute privileges, restrict group members to read and execute, and give only execute privileges to all other users on the system. Only the owner of a file (or the superuser) can change its read, write, and execute privileges.

chown The chown command changes file ownership. In order to change the owner, you must own the file or have superuser privileges. Special group privileges determines a group's ability to use the chown command on files not

owned by the user. The `setprivgrp` command controls special group privileges. To use the `setprivgrp` command, see the "Displaying/assigning special group privileges using SPP-UX commands" section on page 108.

`chgrp` The `chgrp` command changes file group ownership. In order to change the group, you must own the file or have superuser privileges.

See the `chmod(1)`, `chown(1)`, and `chgrp(1)` man pages for additional information.

Default file permissions are assigned by the system whenever you create a new file or directory, and these are governed by your `umask` setting. Unless set up otherwise by you or your system administrator, your default `umask` setting will be 0, which means that new files you create will have read/write permission for everyone (666 or -rw-rw-rw-) and new directories you create will have read/write/search permission for everyone (777 or drwxrwxrwx).

See also the `ll(1)`, `setprivgrp(1M)`, and `umask(1)` man pages.

Access control lists

Access control lists (ACLs) offer a finer degree of file protection than traditional file-mode protection. With ACLs, you can allow or restrict file access to individual users, unrelated to what group the users belong to, with the `chacl` command. Only the owner of a file (or the superuser) can create ACLs with the `chacl` command.

Since you can use both the `chmod` and the `chacl` commands to change access permissions, you need to be aware of how the two commands interact.

- The `chacl` command is a superset of the `chmod` command. Any specific permissions you assign with the `chacl` command are added to the more general permissions assigned with the `chmod` command. For example, suppose you use the chmod command to allow only yourself write permission to `myfile`. You can use the chacl

command to make an exception and allow your manager write permission to `myfile` also. Users other than yourself and your manager will still be denied write permission as previously specified by the `chmod` command.

- Use `chmod` with the `-A` option when working with files that have additional permissions assigned with the `chacl` command. The additional permissions will be deleted if you fail to use the `-A` option with `chmod`.

For additional ACL information see the `lsacl(1)`, `chacl(1)`, and `acl(5)` man pages.

Controlling run-levels

A *run-level* is an SPP-UX state of operation in which a specific set of processes (and their offspring) are permitted to run. This set of processes is defined for each run-level in a file called `/etc/inittab`.

Run-level 2 is the normal operating mode (often called *multi-user mode*). Users must log in to the system in order to gain access. Special processes called `gettys` run in this mode and post the login prompt on your system's terminals. When users log in to your system, the gettys initiate other processes (usually SPP-UX shells) which in turn allow still other processes to be executed as users enter commands.

A special run-level called run-level "s" is also defined. Run-level "s" is a special system administration mode, called *single-user mode*. It is used for performing special tasks where it is desirable to have no one else on the system.

Run levels s and 2 are predefined. You can create new run-levels or change which processes can run at these predefined run-levels, if your needs require. You can define up to six run-levels (1-6). Most systems do not need to define additional run-levels, and modifications to the predefined run-level 2 are usually done to allow `getty` processes to run on terminals being added to a system.

To create a new run-level, make (or change) entries in the `/etc/inittab` file that define how you want the system to operate when the system is in that

run-level. For information on the `/etc/inittab` file, see the `inittab(4)` man page.

When you use SAM to add terminals to your system, SAM makes the entries in the file `/etc/inittab` for you.

Only the superuser can use the `init` command, which changes the system from one run level to another, but anyone having write permission to the file `/etc/inittab` can create new run-levels or redefine existing run-levels.

To protect your system from tampering, ensure that the permissions (and ownership) for the files `/etc/init` and `/etc/inittab` are:

```
-r-xr-xr-x root other /etc/init
-r--r--r-- root root /etc/inittab
```

See the "Creating a new run-level using SPP-UX commands" section on page 112, the "Changing system run-levels using SPP-UX commands" section on page 113, the "Entering the system administration run-level" section on page 114, and the "Returning from the system administration run-level" section on page 115 for specific instructions.

Managing user accounts and group tasks

There are two ways to perform user account and user group tasks on your system:

- Using the System Administration Manager (SAM)
- Using SPP-UX Commands (editing a series of files and creating user directories)

Generally, you should use the SAM method because it is simpler and faster than performing the task with commands. For information about running SAM and navigating within SAM, see Chapter 1.

SAM allows you to control access to your system through its menu-selection and data-entry screens. By combining multiple "manual commands" into single tasks, SAM can save you time and keystrokes. SAM also eliminates the need to know command names and options.

Although SPP-UX commands require you to learn more details than SAM does, you might need or prefer to use SPP-UX commands, for the following reasons:

- SPP-UX commands give you a greater degree of control.
- SAM might not be configured into your system. You have to use SPP-UX commands.
- You might be more comfortable using SPP-UX commands.

If you use SAM to add, remove, or modify users accounts, SAM will edit `/etc/group` for you. You can also use SAM to work directly with groups (add a new group, remove a group, change the list of users in a group). SAM will not edit `/etc/logingroup` explicitly. If `/etc/group` and `/etc/logingroup` are linked together, then when SAM edits `/etc/group` the changes will also be reflected in the `/etc/logingroup` file.

The following are tasks covered in this section:

- Adding a User
- Removing a User
- Customizing the SAM 'Adding and Removing a User' Capabilities
- Deactivating a User's Account
- Reactivating a User's Account
- Displaying/Modifying a User's Account Information
- Adding a Group
- Removing a Group
- Changing a User's Primary Group
- Adding Users to Groups
- Removing Users From Groups
- Displaying/Assigning Special Group Privileges

Each task has an ordered list of instructions, an area for additional information if necessary, and specific examples. In some of the SPP-UX commands method examples the `xargs` command is used with the find command. Output from find is piped to

`xargs` instead of using the `-exec` *primary* option to the find command. This is because when a large number of files or directories are to be processed by a single command, the `-exec` *primary* spawns a separate process for each file or directory, whereas `xargs` collects filenames or directory names into multiple arguments to a single `chgrp` or `chown` command, resulting in fewer processes and greater system efficiency. Specify the full pathname to the command following `xargs` to guarantee expected command behavior. See the `find(1)` and `xargs(1)` man pages for additional information.

Adding a user using SAM

To add a user to your system:

Step 1 Ensure that you have superuser capabilities.

Step 2 Run SAM; type:

```
/usr/bin/sam
```

See Chapter 1 for additional information about using SAM.

Step 3 Select `Users and Groups`, then press the `Open` control button.

Step 4 Select `Users`, and press the `Open` control button.

Step 5 Select `Add ...` from the `Actions` menu.

Step 6 Fill in the `Add a User Account` window fields and press the `Apply` control button.

Step 7 After reading the messages, press the `OK` control button.

To return to the functional area list or functional subarea, choose `Exit` from the `List` menu.

Additional task information

SAM provides an on-line help system to assist you when you need additional information.

Pressing the Help button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the F1 key gives you context-sensitive information for the object at the location of the cursor.

Refer to "Customizing the SAM 'Adding and Removing a User' capabilities" section on page 68" for specific SAM customization instructions.

Even though the user should be unique, there are a few circumstances where it is useful to have several `/etc/passwd` entries with the same user_ID number. For example, consider the following three `/etc/passwd` entries:

```
root:9WsbljlTvWbbw:0:3:Root User Account:/:y/bin/sh
croot:NPt3HW.jBpVz2:0:3:Root User Account (C-Shell):/:/bin/csh
root:dGJbw/DBeDLdo:0:3:Root User Account (K-Shell):/:/bin/ksh
```

On the system with these entries in the `passwd` file, there are three separate accounts (`root`, `croot`, `kroot`) which have superuser capability. Depending on which one is used, the superuser will start up in either the Bourne Shell, the C Shell or the Korn shell.

Because all three accounts have the user_ID "0", the system views all three as the same user. When the system checks to see which user "owns" a file, it compares the "user_ID" associated with the file against the user_ID entries in the `/etc/passwd` file. When it does so, it scans the `/etc/passwd` file from beginning until it finds a user_ID match. This is why files created by the users `croot` and `kroot` (in the above example) will be listed (in the output of the ll command) as being owned by the user root.

Removing a user using SAM

To remove a user from your system using SAM:

Step 1 Ensure that you have superuser capabilities.

Step 2 Run SAM; type:

```
/usr/bin/sam
```

See Chapter 1 for additional information about using SAM.

Step 3 Select `Users and Groups->` and press the `Open` control button.

Step 4 Select `Users` and press the `Open` control button.

Step 5 Choose `Remove ...` from the `Actions` menu.

Step 6 Turn on the check box associated with the action regarding the user's files and press the `OK` control button.

Step 7 After reading the messages, press the `OK` control button.

To return to the functional area list or functional subarea, choose `Exit` from the `List` menu.

Additional Task Information

SAM provides an on-line help system to assist you when you need additional information.

Activating the Help button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the F1 key gives you

context-sensitive information for the object at the location of the cursor.

Note

SAM views a user as a specific user_ID (as opposed to a specific login name) and will not remove any user with the same user_ID as the superuser (user_ID 0). You should never remove the user called "root" from your system. If you have other superusers (users with the user_ID of zero) on your system you can remove them by simply removing their entry from the /etc/passwd file.

If SAM detects that another user has the same UID (user_ID), SAM does not remove the user's files.

SAM will not remove system directories, even if they are owned by a given user. SAM will not remove files across NFS mounts.

SAM updates the /etc/passwd and /etc/group files, but does not update the /etc/logingroup file. If you use /etc/logingroup, edit the file to remove the user from all group entries.

See the "Customizing the SAM 'Adding and Removing a User' capabilities" section on page 68 for specific SAM customization instructions.

Customizing the SAM 'Adding and Removing a User' capabilities

To customize the procedure for adding and/or removing a user:

Step 1 Ensure that you have superuser capabilities.

Step 2 Run SAM; type:

```
/usr/bin/sam
```

See Chapter 1 for additional information about using SAM.

Step 3 Select `Users and Groups->` and press the `Open` control button.

Step 4 Select `Users` and press the `Open` control button.

Step 5 Choose `Task Customization ...` from the `Actions` menu.

Step 6 Fill in the script file name to be executed in one or a combination of the before/after adding/removing a user fields.

Step 7 Press the OK control button.

Step 8 After reading the messages, press the OK control button.

Additional task information

SAM provides an on-line help system to assist you when you need additional information.

Activating the Help button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the F1 key gives you context-sensitive information for the object at the location of the cursor.

There are often additional steps (specific to your operations) that you may want to perform whenever you add or remove a user to your system. SAM allows you to set up shell scripts or executable programs which it will run for you before adding or removing the user, after adding or removing the user, or both before and after adding or removing the user.

There are strict permission and ownership requirements that must be followed for your custom script/program:

1. The file must be owned by root (group ownership not checked).

   Acceptable:

```
-r-xr--r-- 1 root bin 994 May 3 07:44 ct_addnode.ex
-r-xr--r-- 1 root other 994 May 3 07:44 ct_addnode.ex
```

Unacceptable (areas that are highlighted):

```
-r-xr--r-- 1 bin bin 994 May 3 07:44 ct_addnode.ex
-r-xr--r-- 1 joe bin 994 May 3 07:44 ct_addnode.ex
```

2. The file must be writable and executable only by root (note that the file does not have to be writable, but if it is, it can only be writable by root).

   Acceptable:

```
-rwxr--r-- 1 root bin 994 May 3 07:44 ct_addnode.ex
-r-xr--r-- 1 root bin 994 May 3 07:44 ct_addnode.ex
-rwxr----- 1 root bin 994 May 3 07:44 ct_addnode.ex
-r-x---r-- 1 root bin 994 May 3 07:44 ct_addnode.ex
-r-x------ 1 root bin 994 May 3 07:44 ct_addnode.ex
```

   Unacceptable (areas that are highlighted):

```
-rwxrw-rw- 1 root bin 994 May 3 07:44 ct_addnode.ex
-rwxrw-r-- 1 root bin 994 May 3 07:44 ct_addnode.ex
-r-xr-xr-x 1 root bin 994 May 3 07:44 ct_addnode.ex
-rwxr-xr-- 1 root bin 994 May 3 07:44 ct_addnode.ex
```

3. The file must reside in a directory path where all directories (that is, each directory in the directory path) are writable only by owner.

   Suppose the custom command lies in directory `/usr/local/bin`. To successfully pass the validation, the permissions on `/usr`, `/usr/local`, and `/usr/local/bin` must all be "drwxr-xr-x". The permissions cannot be "drwxrwxr-x" or "drwxrwxrwx" (must be writable only by owner). This is typically a problem because `/usr/local` and `/usr/local/bin` are installed with permissions "drwxrwxrwx".

   This means that the system administrators must take care in locating a directory (path) that meets the above requirements (`/usr`, `/usr/bin`, `/usr/sam`, `/usr/sam/bin`, `/usr/sam/config` are just a few examples of directories that at least meet the criteria when installed), or make one of their own that meets the requirements.

These restrictions are only applied at the time of SAM field validation of the command. Once SAM has registered a custom command to be used, the restrictions above are no longer checked by SAM for that command unless the user alters the custom script/program with SAM.

Deactivating a user's account using SAM

To deactivate a user's account:

Step 1 Ensure that you have superuser capabilities.

Step 2 Run SAM; type:

```
/usr/bin/sam
```

See Chapter 1 for additional information about using SAM.

Step 3 Select `Users and Groups->` and press the `Open` control button.

Step 4 Select `Users` and press the `Open` control button.

Step 5 Select the user entry in the object list you wish to deactivate.

Step 6 Choose `Deactivate...` from the `Actions` menu.

Step 7 Turn on the check box for the action regarding the user's files and press the `OK` control button.

Step 8 After reading the messages, press the `OK` control button.

To return to the functional area list or functional subarea, choose `Exit` from the `List` menu.

Additional task information

SAM provides an on-line help system to assist you when you need additional information.

Activating the `Help` button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area

- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the F1 key gives you context-sensitive information for the object at the location of the cursor.

Sometimes it is useful to temporarily suspend a user's ability to log into your system (such as when the user will be away for an extended period of time). The user's files remain on the system and intact, ready for the user when they return and you reactivate their account.

Deactivating a user's account means to make it so that no login password is valid.

Reactivating a user's account Using SAM

To reactivate a user's account:

Step 1 Ensure that you have superuser capabilities.

Step 2 Run SAM; type:

```
/usr/bin/sam
```

See Chapter 1 for additional information about using SAM.

Step 3 Select `Users and Groups->` and press the `Open` control button.

Step 4 Select `Users` and press the `Open` control button.

Step 5 Select the user entry in the object list you wish to reactivate.

Step 6 Choose `Reactivate...` from the `Actions` menu.

Step 7 Optionally fill in a password for the user and press the `OK` control button.

Step 8 After reading the messages, press the `OK` control button.

To return to the functional area list or functional subarea, choose `Exit` from the `List` menu.

Additional task information

SAM provides an on-line help system to assist you when you need additional information.

Activating the Help button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the F1 key gives you context-sensitive information for the object at the location of the cursor.

Displaying/modifying a user's account information using SAM

To display or modify a user's account information:

Step 1 Ensure that you have superuser capabilities.

Step 2 Run SAM; type:

```
/usr/bin/sam
```

See Chapter 1 for additional information about using SAM.

Step 3 Select `Users and Groups->` and press the `Open` control button.

Step 4 Select `Users` and press the `Open` control button.

Step 5 Select the user in the object list.

Step 6 Choose `Modify...` from the `Actions` menu.

Step 7 To view the user's information, press the `Cancel` control button after gathering the information you need.

To modify the user's information, fill in the new information in the `Modify a User` window and press the `OK` control button. After reading the messages, press the `OK` control button. The following user information can be modified:

- login name (user_name)
- password
- user identification number (user_ID)
- primary group identification number (group_ID)
- comment
- login directory
- start up program

To return to the functional area list or functional subarea, choose `Exit` from the `List` menu.

Additional Task Information

SAM provides an on-line help system to assist you when you need additional information.

Activating the `Help` button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the F1 key gives you context-sensitive information for the object at the location of the cursor.

Adding a group using SAM

Step 1 Ensure that you have superuser capabilities.

Step 2 Run SAM; type:

```
/usr/bin/sam
```

See Chapter 1 for additional information about using SAM.

Step 3 Select `Users and Groups->` and press the `Open` control button.

Step 4 Select `Groups` and press the `Open` control button.

Step 5 Choose `Add...` from the `Actions` menu.

Step 6 Fill in the new group name and optionally select the users to be members of the newly created group.

Step 7 Press the OK control button if this is the only group you are adding. Otherwise, press the `Apply` and subsequent `OK` control buttons to return to the `Add a Group` window.

To return to the functional area list or functional subarea, choose Exit from the `List` menu.

Additional Task Information

SAM provides an on-line help system to assist you when you need additional information.

Activating the Help button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the F1 key gives you context-sensitive information for the object at the location of the cursor.

Removing a group using SAM

To remove a group:

Step 1 Ensure that you have superuser capabilities.

Step 2 Run SAM; type:

```
/usr/bin/sam
```

See Chapter 1 for additional information about using SAM.

Step 3 Select `Users and Groups->` and press the `Open` control button.

Step 4 Select `Groups` and press the `Open` control button.

Step 5 Choose `Remove ...` from the `Actions` menu.

You can assign the group's files or another group if desired. Otherwise, SAM will reassign the group's files to the primary group of each of the file's owner.

Step 6 After reading the messages, press the OK control button.

To return to the functional area list or functional subarea, choose `Exit` from the `List` menu.

Additional task information

If the group SAM is removing is a user's primary group, SAM displays an error message and does not remove the group.

SAM provides an on-line help system to assist you when you need additional information.

Activating the `Help` button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system

- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the F1 key gives you context-sensitive information for the object at the location of the cursor.

See the "Customizing the SAM 'Adding and Removing a User' capabilities" section on page 68 for specific SAM customization instructions.

Adding and removing users from groups using SAM

To modify a group's membership list:

Step 1 Ensure that you have superuser capabilities.

Step 2 Run SAM; type:

```
/usr/bin/sam
```

See Chapter 1 for additional information about using SAM.

Step 3 Select `Users and Groups->` and press the `Open` control button.

Step 4 Select `Groups` and press the `Open` control button.

Step 5 Choose `Modify...` from the `Actions` menu.

Step 6 Follow the instructions displayed in the `Modify a Group` window and to add and remove members to and from a group.

Step 7 Press the `OK` control button.

To return to the functional area list or functional subarea, choose Exit from the `List` menu.

Additional Task Information

SAM provides an on-line help system to assist you when you need additional information.

Activating the `Help` button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the F1 key gives you context-sensitive information for the object at the location of the cursor.

Adding a user using SPP-UX commands

Generally you should use the SAM method because it is simpler and faster than performing the task with commands. For information about running SAM and navigating within SAM, see Chapter 1.

To add a user to your system using SPP-UX commands:

Step 1 Ensure that you have superuser capabilities.

Step 2 Make a copy of the `/etc/passwd` file so that it is easy to undo any mistakes that you might make:

```
cp /etc/passwd /etc/passwd.old
```

If you need to undo a mistake later, you can restore the old contents of the file using the command:

```
cp /etc/passwd.old /etc/passwd
```

Step 3 Create an entry in the file `/etc/passwd` for the new user using the text editor of your choice. HP recommends that you use the `vipw` command to ensure that you have exclusive access to the `/etc/passwd` file. The `/etc/vipw` command requires the EDITOR environment variable set to `vi`. Each user attribute must be colon-separated with no leading or trailing spaces. The one-line entry must be in the form:

user_name:password:UID:GID:comment:login_directory:start_up_program

where:

user_name

This is the user's login name (the one that the user will enter at the login: prompt). The login name must have the following characteristics:

- It must begin with an alphabetic character.
- It can include up to eight alphanumeric characters.
- It cannot contain blank spaces.
- It cannot already exist on the system.

password

To ensure the user sets a password when they log in for the first time, place the characters ",.." in this field. For example:

```
bhewlett:,..:567:40:Bill Hewlett:/users/bhewlett:/bin/csh
```

Note

Putting an unencrypted password in the password field will not work. For example, if you want to assign a user the password "secret", the following entry will not allow the user to log in using the password "secret":

```
bhewlett:secret:567:40:Bill Hewlett:/users/bhewlett:/bin/csh
```

To set the password, use the /bin/passwd command.

To leave the new user's account without a password, do not put any characters between the colon (":") separators. For example:

```
dpackard123:40:David Packard:/users/dpackard:/bin/csh
```

Caution

Leaving an account unprotected (without a password), even for a short period of time, is a security risk. If you entered ",.." in the password field, have the user log in as soon as possible to set a password for the account.

The `passwd` command is used to set or change the user's password.

UID

The UID (*user_ID*) is a unique integer value that the system uses to identify the user. If the *user_ID* is 0 (zero), then that user has superuser capabilities. When the system was shipped to you, the *user_ID* "0" was associated with the user root. By convention, the values 1 through 99 are reserved for system use. Therefore, when you are adding a new user to your system, pick for them any unused

number greater than 99 (but less than 60000) for this field. *user_ID*s greater than 59999 are invalid.

GID

This value is the user's primary group GID (*group_ID*) as defined in the third field of the user's primary group entry in the `/etc/group` file. The *group_ID* is an integer value shared by all members of the same group. See "Adding a group using SPP-UX commands" section on page 98 for a description of the `/etc/group` and `/etc/logingroup` file formats.

comment

The *comment* field is used to log information about the identity of the user (or to identify this entry). Although this field is "free-format", using the following comma-separated subfield format is recommended:

```
User's Full Name, Office Location, Office Phone, Home Phone
```

login_directory

This is the absolute pathname to the directory that the user will be placed in when they first log in to the system. The directory need not exist when the entry to `/etc/passwd` is made. However, the directory must exist before the user can log in. This field can be no longer than 63 characters.

start_up_program

This field contains the name of a single command to be executed for the user when he or she logs in; it should be an absolute pathname (for the command). This field can be no longer than 44 characters. Typically this is the name of a shell (`/bin/sh`, `/bin/csh`, `/bin/ksh`, etc.). However, the name can be that of any executable program or command. The command can be either a compiled program or a shell script, but no arguments to the command or script should be supplied. If the command field is left blank, `/bin/sh` is executed by default.

When the user logs in, the command listed in this field is executed and control is passed to that program. Once the program terminates, the user is logged out.

Step 4 Create a login ("home") directory for the user with the `mkdir` command:

```
/bin/mkdir [-p] [-m mode] directory
```

where:

`-p`	specifies intermediate directories are created as necessary. Otherwise, the full path prefix of directory must already exist. The mkdir command requires write permission in the parent directory.
`-m` *mode*	specifies creating the directory as specified by *directory* with the file permissions are set to *mode*, which is a symbolic mode string.
directory	specifies the user login directory.

The login directory is the directory that the users are first placed in when they log in to the system.

The login directory that is defined for a user in the `/etc/passwd` file must exist when the user logs into the system or the login attempt will fail.

It is not necessary for each user to have a separate login directory, but this is how systems are usually set up. It is easier to keep the files of the various users separated if each has their own login directory. This, in turn, makes it easier to work with a given user's files (for example when doing backups, determining how much disk space a given user is using, etc.).

Step 5 Use the pwck command to verify that the `/etc/passwd` file has valid entries:

```
/etc/pwck
```

The pwck command validates the number of fields, login name, user ID, group ID, and whether the login directory and optional program name exist. See the `pwck(1M)` man page for additional information.

Step 6 Create login (shell initialization) files for the user.

In most cases, the start_up_program for a user will be one of the SPP-UX shells (`/bin/sh`, `/bin/csh`, `/bin/ksh`, `/bin/rsh`, etc.). Each of these shells has a set of initialization files that they read when they begin executing as a user is logging in to the system.

Table 1 lists the initialization file executed for each shell in the order they are executed.

Table 1 SPP-UX shell initialization files

shell	initialization files executed at login
/bin/ksh /bin/rksh /bin/rsh /bin/sh	/etc/profile $HOME/.profile
/bin/csh /bin/tcsh	/etc/csh.login $HOME/.login $HOME/.cshrc

The `/etc/profile` and `/etc/csh.login` files contain global instructions/commands that you want executed for every user who logs into the system. These files are in the `/etc` directory so that they are accessible to all users. You should not copy them to each user's directory. The local initialization files are located in the user's login directory ($HOME). These local files typically contain shell commands and environment variable definitions that customize the user's environment and/or automatically run one or more programs for the user.

If the local initialization files exist in a user's directory, the shell attempts to execute the commands in the local files after completing the global files, but before the user receives the first shell prompt.

Examples of the local initialization files are located in the `/etc` directory. Their names begin with the characters "d." (for example, `d.profile`). You may copy these files to a user's login directory and customize them. When you copy these file to the user's login directory, rename the files without the d prefix.

Step 7 Create or customize other initialization files for the user.

Other programs such as the `vi` editor and the various mail programs (`mail`, `elm`, `mailx`) have

initialization files which you may also want to set up for the user.

You may need to change the access permissions of particular files within the user's login directory. See the `ll(1)`, `chmod(1)`, `chown(1)`, `chgrp(1)`, and `umask(1)` man pages for additional information.

Step 8 Change the file ownership and group ownership of the new user's home directory and the files to the user's login name and primary group with the `chown` and `chgrp` commands respectively. You must change the ownership and group ownership of these new files to those of your new user so that the new user can access them.

Change the file ownership of a file or directory with the `chown` command:

```
/bin/chown [-R] new_owner login_dir
```

where:

`-R`	specifies to recursively change the file ownership to *new_owner*. For each *login_dir*, the owner of the directory and all files and subdirectories in the file hierarchy below it are changed to *new_owner*.
new_owner	specifies the login name of the new user.
login_dir	specifies the login directory of the new user.

Change the group ownership of a file or directory with the chgrp command:

```
/bin/chgrp [-R] new_group login_dir
```

where:

`-R`	specifies to recursively change the group ownership to *new_group*. For each *login_dir*, the group of the directory and all files and subdirectories in the file hierarchy below it are changed to *new_group*.
new_group	specifies the primary group of the new user.

login_dir specifies the login directory of the new user.

Step 9 Edit the `/etc/group` file, and optionally, the `/etc/login` group file to add the user's user_name to the names in the comma-separated list of members for the group(s). If you want the user to be able to access files belonging to another group other than their primary group without using the `chgrp` command, edit the `/etc/logingroup` file to add the user to all necessary groups.

A blank line in the `/etc/group` or `/etc/logingroup` file is not allowed. If a blank line appears in the files, all entries after the blank line are ignored. See for specific instructions on editing the `/etc/group` and `/etc/logingroup` files.

A group can have a maximum limit of 200 users.

Use the `grpck` command to check for inconsistencies and verify all entries in the `/etc/group` and `/etc/logingroup` files. This verification includes a check of the number of fields, group name, group ID, and whether all login names appear in the password file. The `grpck` command has the following format:

```
/etc/grpck [group_file]
```

where:

group_file is the name of the group file to be checked. The default group file is `/etc/group`.

Step 10 Have the new user log into the system so that you can verify that you have set up their environment correctly.

Additional task information

Refer to for more information about the vipw command.

If the comment field information in the /etc/passwd file is entered in a comma-separated sub-field format, you can use the `/usr/bin/finger` command to display this information. If you need to modify the user's

comment field information, you can modify the /etc/passwd file directly with the vipw command or you can use the /usr/bin/chfn command. The information in the comment field is referred to as "gecos" information. Refer to the section of this chapter and/or the finger(1) and chfn(1) man pages for additional information. Even though the user should be unique, there are a few circumstances where it is useful to have several /etc/passwd entries with the same user_ID number. For example, consider the following three /etc/passwd entries:

```
root:9WsbljlTvWbbw:0:3:Root User Account:/:/bin/sh
croot:NPt3HW.jBpVz2:0:3:Root User Account (C-Shell):/:/bin/csh
kroot:dGJbw/DBeDLdo:0:3:Root User Account (K-Shell):/:/bin/ksh
```

On the system with these entries in the passwd file, there are three separate accounts (root, croot, kroot) which have superuser capability. Depending on which one is used, the superuser will start up in either the Bourne Shell, the C Shell or the Korn shell.

Because all three accounts have the user_ID "0", the system views all three as the same user. When the system checks to see which user owns a file, it compares the user_ID associated with the file against the user_ID entries in the /etc/passwd file. When it does so, it scans the /etc/passwd file from beginning until it finds a user_ID match. Files created by the users croot and kroot (in the above example) will be listed (in the output of the ll command) as being owned by the user root because root is the first of the three entries.

If you have several users sharing a login directory, the ownerships and permissions of the shell local initialization files may need to be adjusted so that all users sharing that login directory have read access to the local initialization files. There are several ways to do this. One way is to assign one of the users sharing the login directory to be the owner of the files, have all of the users sharing the directory be members of the same group, and give the group members read access to the files. For additional information on file protection, see the ll(1), chmod(1), chown(1), chgrp(1), setprivgrp(1M), and umask(1) man pages.

Another way is to create ACLs (Access Control Lists) for the startup files. ACLs allow access control at the user level versus the group level. For additional ACL information see the `lsacl(1)`, `chacl(1)`, and `acl(5)` man pages.

Examples

To add new user accounts for john, patrickd, and naomil to the system:

Step 1 Login as root.

Step 2 Make a backup copy of the `/etc/passwd` file.

```
cp /etc/passwd /etc/passwd.old
```

Step 3 Update the `/etc/passwd` file to include the users' entries. Convex recommends using the `/etc/vipw` command. The `/etc/vipw` command requires the EDITOR environment variable set to `vi`.

To set the Korn and Bourne shell EDITOR environment variable, type:

```
export EDITOR=vi
```

To set the C shell EDITOR environment variable, type:

```
setenv EDITOR vi
```

Add the following entries to the `/etc/passwd` file:

```
john:,..:342:20:John Smith, 125 Elm Street, 555-2324:
/users/john:/bin/ksh
naomil:,..:1667:20:Naomi Adams,540 Market Ave, 555-9078:
/users/naomil:/bin/ksh
patrickd:,..:24:Patrick Daly,421 Orange Road, 555-6140:
/users/patrickd:/bin/ksh
```

Note the primary group for john and naomil is users while the primary group for patrickd is pub.

Step 4 Create a login directory for each of the new users:

```
mkdir -p /users/john
mkdir -p /users/patrickd
mkdir -p /users/naomil
```

Step 5 Check the /etc/passwd file format:

```
pwck
```

Copy local initialization files to each user's login directory:

```
cp /etc/d.profile /users/john/.profile
cp /etc/d.profile /users/patrickd/.profile
cp /etc/d.profile /users/naomil/.profile
```

Step 6 Create or customize initialization files for the users.

Step 7 Change the ownership and permissions of all of the files and subdirectories created in the login directories for john, patrickd, and naomil using the chown and chgrp commands with the -R option:

```
chown -R john /users/john
chown -R patrickd /users/patrickd
chown -R naomil /users/naomil
chgrp -R users /users/john
chgrp -R pub /users/patrickd
chgrp -R users /users/naomil
```

Step 8 Check that ownership and permissions are correct in the new login directories use the ll command:

```
ll /users
drwxr-xr-x 17 john users 2048 Feb 6 11:18 john
drwxr-xr-x 7 naomil users 3072 Feb 5 15:57 naomil
drwxr-xr-x 9 patrickd users 5152 Feb 5 18:07 patrick
```

If you need to globally change the default owner, group, or other permissions on the files and subdirectories created for the new users, use the chmod command with the -R option. For example:

```
chmod -R u=rwx,g=x,o= /users/john/
```

or

```
chmod -R 710 /users/john/
```

For additional informatio, see the chown(1), chgrp(1), and chmod(1) man pages.

Step 9 Update the /etc/group file, and optionally, the /etc/logingroup file to add the three users' login

names to each users' primary group member list. Additionally, add naomil to the pub group and patrickd to the users group. For example:

```
root:*:0: other:*:1: bin:*:2: sys:*:3: adm:*:4: daemon:*:5:
mail:*:6:lp:*:7:users:*:20:john,naomil,patrickd,kerschen,
michelem,dennism,pvallis pub:*:24:patrickd,naomil,dennism
```

A blank line in the `/etc/group file` is not allowed. If a blank line appears in the `/etc/group` file, all entries after the blank line are ignored. Users patrickd and naomil may access files belonging to both groups without changing their current group if the `/etc/logingroup` file has entries for both users in both groups. Otherwise, the `chgrp` command will be necessary for naomil and patrickd to access files belonging to another group.

Step 10 Check the `/etc/group` file format:

```
/etc/grpck
```

Step 11 Instruct users patrickd, naomil, and john to log in.

Removing a user using SPP-UX commands

Generally, you should use the SAM method because it is simpler and faster than performing the task with commands. For information about running SAM and navigating within SAM, refer to Chapter 1.

To remove a user from your system using SPP-UX commands:

Step 1 Ensure that you have superuser capabilities.

Step 2 Decide the future of the user's files and directories. Your choices are:

- Remove the user's files and directories from the system
- Assign the user's files and directories to another user
- A combination of the above

Step 3 Use the find command to get a list of the files on your system which are owned by the user you are removing:

```
/bin/find / -fsonly hfs -user
user_name -print
```

where:

user_name is the user's login name as defined in the user's `/etc/passwd` file entry.

Files, especially those representing executable programs, can be shared among users in a group or among all of the users of the system. If you decide to remove the user's files, be sure that they will not be needed by other users of your system. See the `find(1)` man page for additional information.

Step 4 Find and remove all ACL (Access Control List) entries for the user. To find and remove all of the ACL entries for the user naomil in the users group type:

```
/bin/find / -fsonly hfs -acl
naomil.users -depth -print | xargs
chacl -d naomil.users
```

For additional ACL information seethe `lsacl(1)`, `chacl(1)`, and `acl(5)` man pages.

Step 5 Search for and remove the user's login name from all entries in the `/etc/group` file. Optionally remove the user's login name from all entries in the `/etc/logingroup` file if it exists.

Use the command `grep` (or the search command in your text editor) to find the entries in `/etc/group` that contain the user_name (login name) belonging to the user you are removing.

Using a text editor, delete the user_name from those entries.

Step 6 Use the `grpck` command to check for inconsistencies and verify all entries in the /etc/group and /etc/logingroup files. This verification includes a check of the number of fields, group name, group ID,

and whether all login names appear in the password file. The `grpck` command has the following format:

```
/etc/grpck [group_file]
```

where:

group_file — is the name of the group file to be checked. The default group file is `/etc/group`.

Step 7 Make a copy of the `/etc/passwd` file so that it is easy to undo any mistakes that you might make:

```
cp /etc/passwd /etc/passwd.old
```

If you need to undo a mistake later, you can restore the old contents of the file using the command:

cp /etc/passwd.old /etc/passwd

Step 8 Remove the user's entry in the /etc/passwd file using the vipw command.

Step 9 Use the pwck command to verify that the `/etc/passwd` file has valid entries:

```
/etc/pwck
```

The `pwck` command validates the number of fields, login name, user ID, group ID, and whether the login directory and optional program name exist. See the `pwck(1M)` man page for additional information.

Additional task information

You should never remove the user called "root" from your system.

Examples

To remove user michelem from the system:

Step 1 Login as root.

Step 2 Change file ownership of all files in the login directory for michelem to rykl and remove all of the files on the rest of the system owned by user michelem.

To change file ownership of all files and directories in the /users/michelem login directory to rykl:

```
xargs chown -R rykl
```

To remove files owned by michelem from the system:

```
xargs rm
```

To remove all of the empty directories owned by user michelem:

```
xargs rmdir
```

Step 3 Remove michelem from any Access Control List entries (ACLs):

```
xargs chacl -d michelem.users
```

Step 4 Locate and remove the login name for the user michelem from all entries in the /etc/group file:

```
grep michelem /etc/group|
```

The output might look like this:

```
photo:*:23:dennisp,janetn,michelem,stevens
therapy:*:23:kimz,michelem,bsmith
database:*:23:michelem,lynnf,rykl
```

After removing michelem from the group member lists, the updated /etc/group file entries should look like this:

```
photo:*:23:dennisp,janetn,stevens therapy:*:23:kimz,bsmith
database:*:23:lynnf,earlg
```

Step 5 Check the /etc/group file format:

```
/etc/grpck
```

Step 6 Make a backup copy of the /etc/passwd file.

```
cp /etc/passwd /etc/passwd.old
```

Step 7 Update the /etc/passwd file to delete the line containing the information for user michelem.

Convex recommends using the /etc/vipw command. The /etc/vipw command requires the EDITOR environment variable set to vi.

To set the Korn and Bourne shell EDITOR environment variable, type:

```
export EDITOR=vi
```

To set the C shell EDITOR environment variable, type:

```
setenv EDITOR vi
```

Check the /etc/passwd file format:

```
pwck
```

Deactivating a user's account using SPP-UX commands

To deactivate a user's account:

Step 1 Ensure that you have superuser capabilities.

Step 2 Make a copy of the /etc/passwd file so that it is easy to undo any mistakes that you might make:

```
cp /etc/passwd /etc/passwd.old
```

If you need to undo a mistake later, you can restore the "old" contents of the file using the command:

cp /etc/passwd.old /etc/passwd

Step 3 Edit the /etc/passwd file using the vipw command:

1. Locate the entry in the /etc/passwd file that corresponds to the user's account that you are planning to deactivate.
2. Replace the encrypted password in the second field with an asterisk "*" (fields are separated by colons ":").
3. Save the /etc/passwd file and exit the editor.

Step 4 Use the pwck command to verify that the /etc/passwd file has valid entries:

```
/etc/pwck
```

The pwck command validates the number of fields, login name, user ID, group ID, and whether the login directory and optional program name exist. See the pwck(1M) man page for additional information.

Additional task information

Caution

You should never deactivate the user called root. This destroys the ability to perform necessary system administration tasks on SPP-UX and usually requires a scratch installation of the operating system.

Deactivating a user's account means to make it so that no login password is valid.

Sometimes it is useful to temporarily suspend a user's ability to log into your system (such as when the user will be away for an extended period of time). The user's files remain on the system and intact, ready for the user when they return and you reactivate their account.

Examples

To deactivate the user account for paul, edit the /etc/passwd file. Convex recommends using the /etc/vipw command. The /etc/vipw command requires the EDITOR environment variable set to vi.

To set the Korn and Bourne shell EDITOR environment variable, type:

```
export EDITOR=vi
```

To set the C shell EDITOR environment variable, type:

```
setenv EDITOR vi
```

The /etc/passwd file entry before deactivating user paul:

```
paul:sIgNXHLuFptVE:209:20:Paul Avonette,Mailstop \
      F13,555-7086, :/users/paul:/bin/ksh
```

The /etc/passwd file entry after deactivating user paul:

```
paul:*:209:20:Paul Avonette,Mailstop \
      F13,555-7086,:/users/paul:/bin/ksh
```

Check the /etc/passwd file format:

```
pwck
```

Reactivating a user's account using SPP-UX commands

To reactivate a user's account:

Step 1 Ensure that you have superuser capabilities.

Step 2 Make a copy of the `/etc/passwd` file so that it is easy to undo any mistakes that you might make:

```
cp /etc/passwd /etc/passwd.old
```

If you need to undo a mistake later, you can restore the old contents of the file using the command:

```
cp /etc/passwd.old /etc/passwd
```

Step 3 Edit the `/etc/passwd` file using the `vipw` command:

1. Locate the entry in the /etc/passwd file that corresponds to the user's account that you are planning to reactivate.
2. Replace the asterisk "*" in the second field of the file with the string ",.." (comma-dot-dot) which forces the user to set a new password for the account the next time he or she logs in.
3. Save the `/etc/passwd` file and exit the editor.

Step 4 Use the `pwck` command to verify that the `/etc/passwd` file has valid entries:

```
/etc/pwck
```

The `pwck` command validates the number of fields, login name, user ID, group ID, and whether the login directory and optional program name exist. See the `pwck(1M)` man page; manual for additional information.

Additional Task Information

Refer to for more information about the vipw command. Examples To reactivate the user account for paul, edit the /etc/passwd file. HP recommends using the /etc/vipw command. The /etc/vipw command requires the EDITOR environment variable set to vi.

To set the Korn and Bourne shell EDITOR environment variable, type:

```
export EDITOR=vi
```

To set the C shell EDITOR environment variable, type:

```
setenv EDITOR vi
```

The /etc/passwd file entry before reactivating user paul:

```
paul:*:209:20:Paul Avonette,Mailstop F13,555-7086, \
      :/users/paul:/bin/ksh
```

The /etc/passwd file entry after reactivating user paul:

```
paul:,..:209:20:Paul Avonette,Mailstop F13,555-7086, \
      :/users/paul:/bin/ksh
```

Check the /etc/passwd file format:

```
pwck
```

Displaying/modifying a user's account information using SPP-UX commands

To display a user's account information:

- Use the finger command to display the user's /etc/passwd file information:

  /usr/bin/finger *user_name*

  where *user_name* is the user's login name as defined in the user's /etc/passwd file entry.

or

- Use the `grep` command to display the user's `/etc/passwd` file information:

  ```
  /bin/grep user_name /etc/passwd
  ```

  where *user_name* is the user's login name as defined in the user's `/etc/passwd` file entry.

To modify a user's account information:

* Ensure that you have superuser capabilities. *Make a copy of the /etc/passwd file so that it is easy to undo any mistakes <idx | file:/etc/passwd | that you might make:

cp /etc/passwd /etc/passwd.old

If you need to undo a mistake later, you can restore the "old" contents of the file using the command:

cp /etc/passwd.old /etc/passwd

* Edit the /etc/passwd file using the vipw or chfn commands to update the following user information:

* login name (user_name) * password * user identification number (user_ID) * primary group identification number (group_ID) * comment * login directory * start up program

For a description of these fields, refer to the section of this chapter.

The chfn command has the following syntax:

/usr/bin/chfn [user_name]

where:

\user_name\ is the user's login name as defined in the user's /etc/passwd file entry.

You must have superuser capabilities to use the chfn command to update other users' account information, but you can change your own account information without superuser capabilities.

* Use the pwck command to verify that the /etc/passwd file has valid entries:

/etc/pwck

The pwck command validates the number of fields, login name, user ID, group ID, and whether the login directory and optional program name exist. See the pwck(1M) man page for additional information.

Additional task information

If you do not have superuser capabilities and you attempt the use the chfn command to update another user's account information, the error message "You are not allowed to change another person's finger entry." is displayed.

Examples

To display account information for user jdoe:

```
finger jdoe
```

```
Login name: jdoe                In real life: John Doe
Bldg: Building 5
Directory: /users/jdoe          Shell: /bin/ksh
On since Feb 10 11:17:04 on pty/ttys5 from mountian.net.ca
2 minutes 25 seconds Idle Time
No Plan.
```

Instead of finger, you can use the grep command:

```
grep jdoe /etc/passwd
```

```
jdoe:QAJZL4Xjg/BMM:1667:20:John Doe,Building 5,555-1234: \
      /users/jdoe:/bin/ksh
```

To update the telephone number for user jdoe with the chfn command:

```
chfn jdoe
```

```
Default values are printed inside of of '[]'.
To accept the default, type .
To have a blank entry, type the word 'none'.

Name [John Doe]:
Location (Ex: 42U-J4) [Building 5]
Office Phone (Ex: 1632) [1234]: 2233
Home Phone (Ex: 5555678) []:
```

Run `pwck` to check your `/etc/passwd` file format.

Adding a group using SPP-UX commands

To add a new group to your system using SPP-UX commands:

Step 1 Ensure that you have superuser capabilities.

Step 2 Make a copy of the `/etc/group` file, and optionally, the `/etc/logingroup` file so that it is easy to undo any mistakes that you might make:

```
cp /etc/group /etc/group.old
cp /etc/logingroup /etc/logingroup.old
```

If you need to undo a mistake later, you can restore the old contents of the files using the commands:

```
cp /etc/group.old /etc/group
cp /etc/logingroup.old /etc/logingroup
```

Step 3 Create an entry for the new group in the `/etc/group` file with the editor of your choice. Optionally create an entry in the `/etc/logingroup` file to allow access to files belonging to other groups without changing the users' effective group. The `/etc/group` and `/etc/logingroup` one-line entries must have the following format:

group_name:group_password:group_ID:user1[,user2][,user3]...

where:

group_name This is the name of your new group. It must begin with an alphabetic character and can include up to 16 alphanumeric characters.

group_password It is recommended that you put an asterisk "*" in this field, which indicates that you will not be using group passwords.

group_ID Like the user_ID field in the file `/etc/passwd`, the group_ID is a unique integer, which is used by SPP-UX to identify the group.

user1,... This is a list of comma-separated user_names (from the first field of the entries in the `/etc/passwd` file).

Step 4 Use the `grpck` command to check for inconsistencies and verify all entries in the `/etc/group` and `/etc/logingroup` files. This verification includes a check of the number of fields, group name, group ID, and whether all login names appear in the password file. The `grpck` command has the following format:

```
/etc/grpck [group_file]
```

where: *group_file* is the name of the group file to be checked. The default group file is `/etc/group`.

Additional task information

The `/etc/group` file is used by the `newgrp` command to check access privileges when a user is attempting to change their effective group. If the user's login name appears in the access list of the group for which access is being requested, the access is granted, thus changing the user's current group to the requested group. The `/etc/logingroup` file, in contrast to `/etc/group`, allows users listed in more than one group access to files belonging to other groups that they are members of without changing their primary or effective groups.

A blank line in the `/etc/group` or `/etc/logingroup` file is not allowed. If a blank line appears in the files, all entries after the blank line are ignored.

A group can have a maximum limit of 200 users.

Optionally, add user entries to Access Control Lists (ACL). For additional ACL information, see the `lsacl(1)`, `chacl(1)`, and `acl(5)` man pages.

Examples

The following listing is a sample /etc/group file:

```
therapy:*:20:dennisp,janetn,jdoe,stevens
photo:*:30:kimz,jdoe,bsmith
leader:*:59:blink,pgomez,stevens
manager:*:67:obones,jab,mlee,fjones
```

To add a user group "users":

Step 1 Login as root.

Step 2 Make a backup copy of /etc/group and /etc/logingroup:

```
cp /etc/group /etc/group.old
cp /etc/logingroup /etc/logingroup.old
```

Step 3 Create an entry in the /etc/group file for user group "users". For example:

```
therapy:*:20:dennisp,janetn,jdoe,stevens
users:*:23:michelem,rykl,karens,starsky,stevens
photo:*:30:kimz,jdoe,bsmith
leader:*:59:blink,pgomez,stevens
manager:*:67:obones,jab,mlee,fjones
```

Step 4 Check the /etc/group file format:

```
/etc/grpck
```

Step 5 Edit the /etc/logingroup file to enable user stevens to access files belonging to groups therapy, leader, and users without changing effective groups:

```
more /etc/logingroup

therapy:*:20:stevens
users:*:23:stevens
leader:*:59:stevens
```

Step 6 Check the /etc/logingroup file format:

```
grpck /etc/logingroup
```

Removing a group using SPP-UX commands

To remove a group from your system:

Step 1 Ensure that you have superuser capabilities.

Step 2 Make a copy of the `/etc/group` file, and optionally the `/etc/logingroup` file, so that it is easy to undo any mistakes that you might make:

```
cp /etc/group /etc/group.old
cp /etc/logingroup /etc/logingroup.old
```

If you need to undo a mistake later, you can restore the old contents of the files using the commands:

```
cp /etc/group.old /etc/group
cp /etc/logingroup.old /etc/logingroup
```

Step 3 Make a copy of the `/etc/passwd` file so that it is easy to undo any mistakes that you might make:

```
cp /etc/passwd /etc/passwd.old
```

If you need to undo a mistake later, you can restore the old contents of the file using the command:

```
cp /etc/passwd.old /etc/passwd
```

Use the `find` command to globally search the system for files with particular file and group ownership:

```
/bin/find / -fsonly hfs -user
user_name -group group_name -depth
-print
```

If you are globally reassigning file or group ownership of all files and directories, use the `find`, `xargs`, and the command to be executed globally (`rm`, `rmdir`, `chown`, `chgrp`, or `chacl`). For example:

```
/bin/find / -fsonly hfs -user
user_name -group group_name -depth
-print | xargs chgrp new_group
```

If you are not globally removing or changing file access permissions, use the find separately from the `rm`, `rmdir`, `chown`, `chgrp`, or `chacl` command.

Step 4 Edit the `/etc/passwd` file to remove user entries if you are removing users from the system.

Step 5 Remove Access Control List (ACL) entries for the group being removed using the `find` and `chacl` commands:

```
/bin/find / -fsonly hfs -acl
.group_name
```

where *group_name* is the name of the group being removed.

Step 6 Use the `pwck` command to verify that the `/etc/passwd` file has valid entries:

```
/etc/pwck
```

The `pwck` command validates the number of fields, login name, user ID, group ID, and whether the login directory and optional program name exist. See the `pwck(1M)` man page for additional information.

Step 7 Edit the `/etc/group` file, and optionally the `/etc/logingroup` file, to remove the group entry for the group you are removing. If users are being removed from the system, remove the user from any other groups.

Step 8 Use the `grpck` command to check for inconsistencies and verify all entries in the `/etc/group` and `/etc/logingroup` files. This verification includes a check of the number of fields, group name, group ID, and whether all login names appear in the `/etc/passwd` file. The `grpck` command has the following format:

```
/etc/grpck [group_file]
```

Additional task information

A blank line in the `/etc/group` or `/etc/logingroup` file is not allowed. If a blank line appears in the files, all entries after the blank line are ignored. Refer to "Adding a user using SPP-UX commands" on page 78 for a description of the `/etc/group` and `/etc/logingroup` files. See also the `group(4)` man page.

For additional ACL information, see the `lsacl(1)`, `chacl(1)`, and `acl(5)` man pages.

Changing a user's primary group using SPP-UX commands

To change a user's primary group:

Step 1 Ensure that you have superuser capabilities.

Step 2 Make a copy of the `/etc/passwd` file so that it is easy to undo any mistakes that you might make:

```
cp /etc/passwd /etc/passwd.old
```

If you need to undo a mistake later, you can restore the old contents of the file using the command:

```
cp /etc/passwd.old /etc/passwd
```

Step 3 Determine the *group_ID* number of your user's new primary group by looking at the new primary group's entry in the /etc/group file. The third field of the group entry in the `/etc/group file` (fields are separated by colons ":") contains the *group_ID*.

Step 4 Edit the `/etc/passwd` file using the `vipw` command to replace the primary group ID, *pri_group_ID* (fourth field), of your user's entry with the new primary group ID.

Step 5 Use the `pwck` command to verify that the `/etc/passwd` file has valid entries:

```
/etc/pwck
```

The `pwck` command validates the number of fields, login name, user ID, group ID, and whether the login directory and optional program name exist. See the `pwck(1M)` man page for additional information.

Step 6 Make a copy of the `/etc/group` file, and optionally the `/etc/logingroup` file, so that it is easy to undo any mistakes that you might make:

```
cp /etc/group /etc/group.old
cp /etc/logingroup /etc/logingroup.old
```

If you need to undo a mistake later, you can restore the old contents of the files using the commands:

```
cp /etc/group.old /etc/group
cp /etc/logingroup.old /etc/logingroup
```

Step 7 Edit the /etc/group file to add the user's login name to the entry which corresponds to their new primary group. If you do not want the user to continue to be a member of their previous primary group, you will also need to remove the user's login name from the list of user members of their previous primary group.

A blank line in the `/etc/group` or `/etc/logingroup` file is not allowed. If a blank line appears in the files, all entries after the blank line are ignored.

A group can have a maximum limit of 200 users.

Step 8 Use the `grpck` command to check for inconsistencies and verify all entries in the `/etc/group` and `/etc/logingroup` files. This verification includes a check of the number of fields, group name, group ID, and whether all login names appear in the `/etc/passwd` file.

The `grpck` command has the following format:

```
/etc/grpck [group_file]
```

where *group_file* is the name of the group file to be checked. The default group file is `/etc/group`.

Additional task information

Optionally, update the Access Control List (ACL) entries to replace, add, or delete the necessary file access permissions. For additional ACL information, see the `lsacl(1)`, `chacl(1)`, and `acl(5)` man pages.

Adding users to groups using SPP-UX commands

To add users to a group (without changing the users' primary group):

Step 1 Ensure that you have superuser capabilities.

Step 2 Make a copy of the `/etc/group` file, and optionally the `/etc/logingroup` file, so that it is easy to undo any mistakes that you might make:

```
cp /etc/group /etc/group.old
cp /etc/logingroup /etc/logingroup.old
```

If you need to undo a mistake later, you can restore the old contents of the files using the commands:

```
cp /etc/group.old /etc/group
cp /etc/logingroup.old /etc/logingroup
```

Step 3 Edit the `/etc/group` file, and optionally the `/etc/logingroup` file, to add the user's *user_name* to the names in the comma-separated list of members for the group(s) for which they are to be members. If you want the user to be able to access files belonging to another group other than their primary group without using the `chgrp` command, edit the `/etc/logingroup` file to add the user to all necessary groups.

Step 4 Use the `grpck` command to check for inconsistencies and verify all entries in the `/etc/group` and `/etc/logingroup` files. This verification includes a check of the number of fields, group name, group ID, and whether all login names appear in the `/etc/passwd` file. The `grpck` command has the following format:

```
/etc/grpck [group_file]
```

where *group_file* is the name of the group file to be checked. The default group file is `/etc/group`.

Additional task information

A blank line in the `/etc/group` or `/etc/logingroup` file is not allowed. If a blank line appears in the files, all entries after the blank line are ignored. Refer to the section of this chapter for a description of the `/etc/group` and `/etc/logingroup` file formats.

A group can have a maximum limit of 200 users.

Optionally, update the Access Control List (ACL) entries to replace, add, or delete the necessary file access permissions. For additional ACL information

see the `lsacl(1)`, `chacl(1)`, and `acl(5)` man pages.

Examples

Here is a sample /etc/group file:

```
cats:*:15:donna,woody
naomil:*:20:mj,michelem,kerschen
```

To add users pixie and pepper to groups "cats" and "naomil":

Step 1 login as root.

Step 2 Make a copy of the `/etc/group` file:

```
cp /etc/group /etc/group.old
```

Optionally, make a copy of the `/etc/logingroup` file. *

Step 3 Edit the `/etc/group` file to add the users pixie and pepper to the group member lists. For example:

```
cats:*:15:donna,woody,pixie,pepper
naomil:*:20:mj,michelem,kerschen,pixie,pepper
```

Optionally edit the `/etc/logingroup` file.

Step 4 Check the `/etc/group` file format:

```
/etc/grpck
```

Optionally run `grpck` on the `/etc/logingroup` file.

Removing users from groups using SPP-UX commands

If you are removing users from the system, see "Removing a user using SPP-UX commands" on page 88 of this chapter. If you are changing the user's primary group, see "Changing a user's primary group using SPP-UX commands" on page 103 of this chapter. Otherwise, this procedure assumes that you are not removing users from their primary groups.

To remove users from groups:

Step 1 Ensure that you have superuser capabilities

Step 2 Make a copy of the `/etc/group` file, and optionally the `/etc/logingroup` file, so that it is easy to undo any mistakes that you might make:

```
cp /etc/group /etc/group.old
cp /etc/logingroup /etc/logingroup.old
```

If you need to undo a mistake later, you can restore the old contents of the files using the commands:

```
cp /etc/group.old /etc/group
cp /etc/logingroup.old /etc/logingroup
```

Step 3 List the files and directories owned by the user using the find command:

```
/bin/find / -fsonly hfs -user
user_name -group group_name -depth
-print
```

where:

user_name is the login name of the user as defined in the `/etc/passwd` file.

group_name is the group from which the user is being removed.

If the list of files is long you can redirect the output to a file or redirect the output to the more command. For example:

```
/bin/find / -fsonly hfs -user
user_name -group group_name -depth
-print | more
```

Step 4 Remove Access Control List (ACL) entries for all users being removed from the group using the `find` and `chacl` commands:

```
/bin/find / -fsonly hfs -acl
user_name.group_name -depth -print
xargs chacl -d user_name.group_name
```

Step 5 Edit the `/etc/group` file, and optionally the `/etc/logingroup` file, to remove the user's *user_name* from the list of members for a group.

Do not remove the user's name from the user's primary group defined in the `/etc/passwd` file unless you are removing the user from the system. If

you want to change the user's primary group, see "Changing a user's primary group using SPP-UX commands" on page 103 of this chapter.

A blank line in the `/etc/group` or `/etc/logingroup` file is not allowed. If a blank line appears in the files, all entries after the blank line are ignored.

Step 6 Use the `grpck` command to check for inconsistencies and verify all entries in the `/etc/group` and `/etc/logingroup` files. This verification includes a check of the number of fields, group name, group ID, and whether all login names appear in the password file. The `grpck` command has the following format:

```
/etc/grpck [group_file]
```

where *group_file* is the name of the group file to be checked. The default group file is `/etc/group`.

Additional task information

For additional ACL information see the `lsacl(1)`, `chacl(1)`, and `acl(5)` man pages.

Displaying/assigning special group privileges using SPP-UX commands

To display special group privileges use the getprivgrp command:

```
/usr/bin/getprivgrp [-g | group_name]
```

where:

`-g` lists access privileges that have been granted to all groups. Otherwise, access privileges are listed for all privileged groups to which the requestor belongs.

group_name is the name of the group as specified in the `/etc/group` file.

If *group_name* is supplied, access privileges are listed for that group only. If the requestor is not a member

of the *group_name* specified, no information is displayed. The superuser is a member of all groups.

To assign special group privileges:

Step 1 Ensure that you have superuser capabilities.

Step 2 Assign special privileges using the `setprivgrp` command. There are three formats for the `setprivgrp` command.

One format of the `setprivgrp` command is:

```
/etc/setprivgrp group_name [privilege]
```

where:

group_name is the name of the group as specified in the `/etc/group` file.

privilege is one or more of the following privileges:

RTPRIO gives access to the `rtprio` command and system call, which allows the setting of real-time priorities.

MLOCK gives access to the `plock` system call, which allows processes to be locked in memory and allows use of the SHM_LOCK command (used with the `shmctl` system call).

CHOWN gives access to the `chown` command and system call, which allows members of the group to change the ownership of files on the system.

LOCKRDONLY gives access to the `lockf` system call to set locks on files that are open for reading only.

SETRUGID gives access to the `setuid` and `setgid` system calls to change, respectively, the real user ID or real group ID of a process.

The other two formats of the setprivgrp command are:

```
/etc/setprivgrp -g [privilege]

/etc/setprivgrp -n [privilege]
```

where:

`-g`	specifies that all groups have access to the specified privilege(s).
`-n`	specifies that no groups have access to the specified privilege(s).
privilege	is one or a combination of the special privileges (RTPRIO, MLOCK, CHOWN, LOCKRDONLY, or SETRUGID).

For additional information see the `rtprio(1)`, `rtprio(2)`, `plock(2)`, `shmctl(2)`, `chown(1)`, `chown(2)`, `lockf(2)`, `setuid(2)`, `setgid(2)`, `setprivgrp(2)`, and `setprivgrp(1M)` man pages.

Capabilities set by this command are not added to existing capabilities for the same group. If you want to add a capability for a particular group, you must respecify all capabilities that were already set for that group as well as the new capability.

Note

Specifying no privileges removes all special privileges for the group.

Additional task information

Any user whose current *group_ID* matches the *group_ID* of a privileged group will have access to the special capabilities assigned to that group. A group can have any one, two or all five of the capabilities associated with it.

Examples

- To set real-time priorities and enable user processes to lock process text and data into memory for the development group:

```
setprivgrp development RTPRIO MLOCK
getprivgrp
global privileges: CHOWN
development: RTPRIO MLOCK
```

- To set real-time priorities and enable user processes to lock process text and data into memory for all groups:

```
setprivgrp -g RTPRIO MLOCK
```

- To deny real-time priorities and disable user processes to lock process text and data into memory for all groups:

```
setprivgrp -n RTPRIO MLOCK
```

For additional information, see the `setprivgrp(1M)` and `getprivgrp(1)` man pages.

Managing run-levels

A run-level is an SPP-UX state of operation in which a specific set of processes (and their offspring) are permitted to run. This set of processes is defined, for each run-level, in a file called `/etc/inittab`.

The following list contains tasks covered in this section:

- Creating a New Run-Level Using SPP-UX Commands
- Changing System Run-Levels Using SPP-UX Commands
- Entering the System Administration Run-Level
- Returning From the System Administration Run-Level

Creating a new run-level using SPP-UX commands

To create new run-levels:

Step 1 Ensure that you have superuser capabilities.

Step 2 Make a copy of the `/etc/inittab` file so that it is easy to undo any mistakes that you might make:

```
cp /etc/inittab /etc/inittab.old
```

If you need to undo a mistake later, you can restore the old contents of the file using the command:

```
cp /etc/inittab.old /etc/inittab
```

Step 3 Edit the `/etc/inittab` file with the editor of your choice to create a one-line entry in `/etc/inittab` with the following format:

id:rstate:action:process

where:

id	is a unique four-character identifier, used to identify an entry.
rstate	is a list of run-levels to which each entry applies.
action	is an action to be performed, such as respawn.
process	is the command that will be executed when that run-level is entered.

See the `init(1M)` and `inittab(4)` man pages for a detailed description of entries in the `/etc/inittab` file.

Step 4 Edit the /etc/inittab file to change the initdefault entry in your test version to "s". By changing the initdefault entry to "s", you will come up in run-level "s" when you boot. You can change to run-level 2 after booting by executing init 2. If your new run-level 2 does not work, you can still reboot.

"s" is not a normal run-level. If you create this test version, you should replace the "s" with "2" after testing is complete. Run-level "s" is for system maintenance only.

If you do not have a working state for the `initdefault` state, you may not be able to boot your system. After thoroughly testing your changes, restore the original `initdefault` value.

Examples

The following is an example `/etc/inittab` for a system that contains a system console and six terminals. The initdefault run-level is run-level 2. Run-level 2 is a multi-user run-level, with a `getty` on every terminal.

```
init:2:initdefault:
stty::sysinit:stty 9600 clocal icanon echo opost onlcr ienqak \
        ixon icrnl ignpar </dev/systty
brc1::bootwait:/etc/bcheckrc &1 # fsck, etc.
slib::bootwait:/etc/recoversl &1 #shared libs
brc2::bootwait:/etc/brc >/dev/console 2>&1 # boottime commands
link::wait:/bin/sh -c "rm -f /dev/syscon; \
         ln /dev/systty /dev/syscon" >/dev/console 2>&1
cwrt::bootwait:cat /etc/copyright >/dev/syscon
        # legal requirements
rc ::wait:/etc/rc &1 # system initialization
powf::powerwait:/etc/powerfail >/dev/console 2>&1
        # power fail routines
lp ::off:nohup sleep 999999999 </dev/lp & stty 9600 </dev/lp
cons:012456:respawn:/etc/getty -h console console
        # system console
t1:2:respawn:/etc/getty tty01 H
t2:2:respawn:/etc/getty tty02 H
t3:2:respawn:/etc/getty tty03 H
t4:2:respawn:/etc/getty tty04 H
t5:2:respawn:/etc/getty tty05 H
t6:2:respawn:/etc/getty tty06 H
```

Changing system run-levels using SPP-UX commands

The following is a general procedure for changing the system from one run-level to another. You must be logged in at the system console as the superuser to change the system's run-level.

Step 1 Warn all users who are currently logged in before you change run-levels.

Changing to another run-level while users are logged on will kill (terminate) their processes if the run-level you are moving to does not contain `rstate` entries in `/etc/inittab` for their terminal. You can use the `write` or `wall` commands to communicate with the users. The `wall` (write all) command immediately sends your message to the terminal of each user on the system.

If each `getty` (terminal) entry has the new run-level in its `rstate` field, or if the `rstate` field is empty (implies all numbered run-levels), you don't need to ask them to log off; their processes will not be killed (unless your new run-level is run-level "s").

Step 2 To change to a run-level other than run-level "s", use the command:

```
/etc/init new_run-level
```

where *new_run-level* is the number of the run-level you want to enter.

To change to run-level "s", use the command:

```
/etc/shutdown
```

Caution

You should not change to run-level "s" without using the `shutdown` command (that is, do not execute `init s`). The `shutdown` command provides safeguards to "cleanly" bring your system to single-user mode (run-level "s").

Run-level 0 is a special run-level reserved for system installation. Do not use run-level 0.

Entering the system administration run-level

Many of the system maintenance tasks you perform as system administrator require the system to be in single-user mode (run-level "s") so that you can ensure that no one else is on the system while you're performing those tasks. In this run-level, the only access to the system is through the system console by the user root, and the only processes running on the system will be the shell on the system console, background daemon processes started by `/etc/rc`, and processes that you invoke. Commands

requiring an inactive system (such as `fsck`) should be run in run-level "s".

Use the shutdown command, instead of init s when changing your system's run-level from any numbered run-level (run-levels 1 through 6) to run-level "s". The shutdown command kills all non-essential processes and brings the system safely to run-level "s" (without leaving system resources in an unusable state).

The shutdown command also allows you to specify a grace period to allow time for your users to terminate their work before the system goes down. The grace period is given (in number of seconds to wait) immediately following the command name. For example:

To enter run-level "s" with a grace period of 30 seconds, type in:

```
/etc/shutdown 30
```

This will automatically warn all users that they have 30 seconds to log off, kill all processes, and safely bring the system to run-level "s".

For details on how to use the `shutdown` command, see Chapter 4; aso see the `shutdown(1M)` pan page.

Returning from the system administration run-level

When you want to change your system's run-level from run-level "s" (single-user mode) to one of the other run-levels, it is best to do so by rebooting your system. You can use the `reboot` command to do this. You can also use the `init` command as described earlier in this chapter (See the "Changing system run-levels using SPP-UX commands" section on page 113) to change to the new run-level.

Managing an SPP-UX file system

6

As a system administrator responsible for managing your SPP-UX file system, you will be performing the following major tasks:

- Creating file systems
- Adding and removing local and remote auxiliary file systems
- Monitoring and controlling the disk space consumed by users' files
- Moving a file system from one disk to another
- Adding or removing swap space in a file system

This chapter describes how to accomplish most of these tasks using SPP-UX commands.

Terms used in this chapter

The following terms appear frequently in this chapter. You can scan the list now and refer to it later.

block device — A hardware device that transmits and receives data in multiple-byte blocks (rather than by streams of individual bytes) or does block-buffered input/output.

block special file — A special file associated with a mass storage device (such as a hard disk or tape cartridge drive) that transfers data in multiple-byte blocks, rather than in a series of individual bytes. See **device file**.

CD-ROM file system — A Read Only Memory file system on Compact Disk. You can read data from a CD-ROM file system, but you cannot write to one.

character special file — A special file associated with I/O devices that transfer data byte-by-byte. Typical byte-mode I/O

devices include printers, nine-track magnetic tape drives, and disk drives when accessed in *raw* mode. Disk drive access via character devices is typically faster than via block devices. Character device file are sometimes called raw device files.

cylinder

On disk drives consisting of several disks, the arrangement of disk tracks under read/write heads that are in the same relative position.

device file

A file used by the computer to communicate with a device. The file tells the operating system the location of the device and what device driver to use. There are block device files (used for transmitting data in multiple-byte blocks) and character device files (used for transmitting data byte-by-byte). Device files are typically stored in the `/dev` directory.

Block device files are stored in the directory `/dev/dsk`.

Character device file are stored in the directory `/dev/rdsk`.

device swap space

A disk or disk section reserved exclusively as swap space.

disk quotas

Disk usage limits that a system administrator can assign to users of a file system.

disk section

A logical division or partition on a hard disk in which a file system or a swap location can be placed. Convex Exemplar systems use disks that are partitioned in numerous sections.

file system

The organization of files on storage devices. The term "file system" can refer either to the entire file system tree or to a subsection of that file system contained on an individual disk, which can be mounted or unmounted from the tree.

fragment

A part of a block. The end of a file that is not a whole block is typically stored as a fragment. The size of a fragment can be specified; the use of a small fragment size adversely affects performance but leaves less wasted space.

HFS file system

A file system in which the files are arranged on a disk within hierarchical directories.

inode	A data structure containing information about a file, such as file type, pointers to data, owner, group, and protection information.
kernel	The actual operating system program that executes and runs, controlling the processes, hardware, file system, and so on.
long file names	File names using more than 14 (but not exceeding 255) characters. Long file names are incompatible with file systems configured for short file names.
mount	To add an auxiliary (removable) file system to an active existing file system.
mount directory	A directory in an existing file system that serves as the root directory (the mount point) of a mounted file system.
mount point	See mount directory.
NFS client	A machine that mounts (via the network) a file system located on a remote NFS server.
NFS file system	A file system accessible over a network via the NFS Services product.
NFS server	A computer with local file systems that are being accessed via the network by remote computers, or NFS clients.
root directory	The highest level directory in a file system. In any mountable file system (any file system other than the root file system) the root directory is the mount directory. The / directory, also known as the root directory, is the highest level in the SPP-UX file system overall; the / file system cannot be unmounted because it contains the running operating system.
root file system	Or root (/); the file system that contains the kernel and other operating system files.
short file names	Files with names consisting of 14 or fewer characters. Short file names are compatible with file systems configured for long file names.
single-user mode	When a computer system is accessible to only one user, usually the system administrator.
swap space	Space on a disk used for storing the process image temporarily.

unmount To remove an auxiliary file system from the existing file system.

Overview of SPP-UX file systems

This section briefly describes the types of file systems you will work with on your computer system.

What does "file system" mean?

In one sense, the word "file system" refers to the entire SPP-UX file system tree, the organization of all files on the system.

The SPP-UX file system is a hierarchical, upside down tree-like structure in which the files, like leaves, are at the bottom or the ends of a branching structure that leads upward through subdirectories to a single root (written) directory.

Mountable file systems

The word "file system" also refers to the specific collection of files on a storage device such as a disk.

You can create the structure for a new file system on a disk by using the `newfs(1M)` command. You can also use SAM to create a file system. Once created, the file system, even though it is empty, encompasses the area on the disk in which it is created.

To use or access the file system, you need to mount that file system to the existing file system tree. Except for the root (/) file system on the system disk, you can mount and unmount all file systems on disks to and from the existing SPP-UX file system tree.

You refer to an auxiliary file system by the name of the device file associated with the disk that contains the file system. You can mount the auxiliary file system by attaching it to a directory (the mount directory) in the root file system. Use the `mount(1M)` command, described later.

You can unmount a file system, too, and then mount it again at a different mount point.

When users traverse the file system, they can move from the / directory to the files in /users/Bob as easily as they can move from / to the files in /usr even though /usr and /users/Bob are on different disks. As a user moves from one part of the SPP-UX file system to another, it isn't apparent that the file system actually consists of separate file systems on different devices.

Listing mounted file systems

To see a list of the file systems mounted on your system, use the bdf(1M) command. For example:

```
bdf

Filesystem kbytes used avail capacity
Mounted on

/dev/dsk/c201d0s0 484960 239992
196472 55% /

/dev/dsk/c201d1s0 237810 47943 166086
22% /users

.

.
```

In the above example listing, the file system on disk with the device file /dev/dsk/c201d1s0 is mounted at the mount directory /users on the root (/) file system in /dev/dsk/c201d0s0.

Types of file systems

The principal types of file systems used by SPP-UX are the HFS, or high performance file system, the NFS, or network file system, and the CDFS, or CD-ROM file system.

HFS file systems. HFS is an acronym for High-performance File System. HFS file systems physically reside on mass storage devices, usually hard disk drives.

NFS file systems. NFS is an acronym for Network File Services. NFS file systems are remote HFS file systems that are accessible over a network that can be used in a local file system.

CD-ROM file systems. CD-ROM is an acronym for Compact Disk Read-Only Memory. The information on the CD is virtually permanent; you can read data from a CD, but you cannot write to one. Data on a CD is prepared and mastered using a specialized publishing process. A CD-ROM file system (CDFS) allows easy retrieval of large amounts of information that requires no modification.

The arrangement of files in a CD-ROM file system is tree-like as in HFS file systems. You can use SPP-UX commands to list, print, or copy files in the CD-ROM file system. However, some commands, such as fsck or mkfs for example, are not supported because of the read-only nature of a CD-ROM file system.

Disk layout

When you add a disk to the system, you can designate how the space on the disk is to be proportioned between file system space and space for swapping . The file `/etc/disktab` shows listing of the various possible layouts available for supported disks. Some examples in this chapter demonstrate the use of `/etc/disktab`.

SPP-UX system files

Many important system files are located in the directories and subdirectories of the root (/) file system, described in

Root file system subdirectories

Table 2

Directory	Contents
`/bin`	Compiled, often-used commands.
`/dev`	Block and character special device files used to communicate with devices. See the `mknod(1M)` man page.
`/etc`	Most system administrator commands and configuration (customization) files.

Table 2

Directory	Contents
/etc/newconfig	Customized configuration files and shell scripts during an update so you can use them for reference. You typically copy many of these files back into /etc. The /etc/newconfig/README file contains useful information about files in /etc/newconfig.
/etc/conf	Kernel configuration description files.
/etc/filesets	A list of loaded filesets.
/lib	Object code libraries and related utilities.
/mnt	Users' home directories (usually).
/system	Revision lists and customize scripts from installation.
/tmp	Temporary files.
/usr	Commands and log files.
/usr/adm	System administration data files.
/usr/bin	Commands not required to boot, restore, or repair the file system.
/usr/contrib	Files and commands contributed by user groups.
/usr/contrib/bin	Contributed commands.
/usr/contrib/lib	Contributed object libraries.
/usr/contrib/man	On-line documentation for contributed software.
/usr/convex	Files and commands for optional Convex software products.
/usr/convex/bin	Commands for optional Convex software products.
/usr/convex/lib	Object libraries for optional Convex software products.
/usr/convex/man	On-line documentation for optional Convex software products.

Table 2

Directory	Contents
/usr/diag	Diagnostic tools.
/usr/include	High-level C language header files; the shared definitions.
/usr/include/local	Site-specific C language header files.
/usr/include/sys	Low-level, kernel-related C language header files.
/usr/lib	Less-used object-code libraries, utilities, lp commands, and miscellaneous data files.
/usr/lib/uucp	Configuration files for UUCP at install.
/usr/local	Localized, site-specific files.
/usr/local/bin	Localized, site-specific commands.
/usr/local/lib	Object code libraries for the site-specific commands.
/usr/local/man	On-line documentation for the site-specific softrware.
/lost+found	Orphaned files and directories created by newfs and used by fsck.
/usr/mail	Used by the mail facilities for your mail box.
/usr/news	System-wide news files.
/usr/spool	Spooled (queued) files for various programs.
/usr/spool/cron	Spooled jobs for cron and at.
/usr/spool/lp	Control and working files for the lp spooler.
/usr/spool/uucp	Queued work files, lock files, log files, status files, and other files for UUCP.
/usr/spool/uucppublic	Files freely accessible to remote systems via LAN and uucp.
/usr/tmp	Temporary large files.
/usr/man	All shipped on-line documentation for SPP-UX.

Table 2

Directory	Contents
/usr/man/cat1...cat9	On-line documentation that has already processed to speed up access.
/usr/man/cat1.Z...cat9.Z	Compressed versions of cat directories.
/usr/man/man1...man9	The unformatted on-line documentation pages.
/usr/man/man1.Z...man9.Z	Compressed versions of the on-line documentation pages.

The diskutil disk utility

The `diskutil` command is used to perform many of the tasks associated with installing and using disks under SPP-UX. It can be used interactively or strictly through the command line. If no command is supplied on the command line, `diskutil` operates as an interactive shell. Otherwise, the single supplied command is executed and diskutil exits.

There are several operational modes you should be aware of. `diskutil` can be run under SPP-UX or HP-UX (to prepare disks for use under SPP-UX). Different techniques must be used internally by `diskutil` to operate on a disk that may be active when running under SPP-UX. `diskutil` attempts to detect this operating mode automatically.

`diskutil` can be used to prepare disks for Exemplar hardware or HP 700 Series workstations. There are some differences in how disks are configured for these to systems. There is not currently a way to determine what system the disk is targeted for, so Exemplar hardware is assumed.

The `diskutil` command has the following format:

```
diskutil [-d disk] [-h | -s] [-H | -S]
[-v] [command]
```

`diskutil` recognizes the following command-line options:

Option Description

`-d` *disk* Equivalent to a 'Select Disk *disk*' command. See below.

`-h` Host operating system is HP-UX.

`-H` Disks are targeted for an HP 700 Series workstation.

`-s` Host operating system is SPP-UX.

`-S` Disks are targeted for Exemplar hardware.

`-v` Verbose mode.

`diskutil` recognizes the following commands:

- `Exit`
- `Help`
- `MAKe`
- `MAP`
- `Prepare`
- `Quit`
- `SELect`
- `SET`
- `SHow`
- `UNMap`
- `UNSet`

diskutil Exit command

This command terminates `diskutil`. There are no parameters for this command.

diskutil Help command

This command prints on-line help for `diskutil`. The `Help` command has the following syntax:

`Help` *command*

Prints on-line help for *command*.

diskutil MAKe Partition command

This command creates or modifies a disk partition. The MAKe command has the following syntax:

```
MAKe Partition partition Size size \ [
Offset offset | After partition | Before
partition ] \ [ Description "string" ]
```

Creates or changes a partition with the specified *size* and optionally sets the description. You must specify either the Offset, After, or Before parameter. When modifying an existing partition, the size or offset can be omitted. Size and offset are in bytes and must be a multiple of 1024 bytes. A 'K' or 'M' suffix indicates kilobytes or megabytes.

diskutil MAP command

This command sets the logical unit name for a disk. The MAP command has the following syntax:

```
MAP Disk To logical-unit
```

```
MAP Disk type node:ctlr:tgt:lun To logical-unit
```

The second form is required for disks with no current map ping which cannot be selected with the select command. The mapping of an active drive cannot be changed. A newly mapped drive must be selected before subsequent operations can be performed on it. Any previous disk selection becomes invalid after this command.

diskutil Prepare Disk command

This command creates the required `lif` data structures on the selected disk to make it usable under SPP-UX. The `Prepare Disk` command has the following syntax:

```
Prepare Disk [ Override ]
```

If the disk already has `lif` data structures, the `override` keyword must be used to re-prepare the disk. Existing data will be lost.

diskutil SELect Disk command

This command selects a disk for further `diskutil` operations. The `SELect Disk` command has the following syntax:

```
SELect Disk disk_name
```

diskutil SELect Volume command

This command selects a lif volume on which to operate. The `SELect Volume` command has the following syntax:

```
SELect Volume lif_volume_name
```

diskutil SET Partition command

This command sets a partition's description or flags field.creates or modifies a disk partition. The `SET Partition` command has the following syntax:

```
SET Partition partition [ Description
"string" ] \ [ Flag { Crashdump |
Default_pager | Vnode_pager } ]
```

diskutil SHow Directory command

This command shows a display of the `lif` directory. The `SHow Directory` command has the following syntax:

```
SHow Directory
```

diskutil SHow Partitions command

This command shows a display of the partition table. The `SHow Partitions` command has the following syntax:

```
SHow Partitions
```

diskutil UNMap Disk command

This command unmaps (removes) the logical unit name mapping for a disk. The `UNMap Disk` command has the following syntax:

```
UNMap Disk [ type:node:ctlr:tgt:lun ]
```

The extended disk specification is required under SPP-UX to prevent the disk from being opened by `diskutil` and thus made active. The mapping of an active drive cannot be removed.

diskutil UNSet Partition command

This command snsets (removes) a partition's description or clear a flag. The `UNSet Partition` command has the following syntax:

```
UNSet Partition partition [ Description
"string" ] \ [ Flag { Crashdump |
Default_pager | Vnode_pager } ]
```

Managing printers 7

What is the Line Printer Spooling System?

The Line Printer Spooling System (`lpss`) is a set of programs, shell scripts, and directories that control your printers and the flow of data going to them. It:

- helps prevent intermixed listings.
- provides control of printout routing.
- allows users to cancel, restart and adjust the priority of print requests.

Once a printer has been added to your system (that is, its driver is in your kernel configuration, and an appropriate device file exists), it can be added to the `lpss` (for example you can redirect the output of a command to the device file associated with the printer). We recommend adding all printers to the `lpss`. If you do not add your printer or plotter to the `lpss`, there is no coordination between multiple users and intermixed listings can occur. Unspooled printing is not recommended. The purpose of the `lpss` is to automatically coordinate between multiple users and prevent intermixed listings.

Note

The term *printer* can be interchanged with the term *plotter* for the tasks described in this chapter.

This chapter will cover the following topics:

- Components of the `lpss`
- Remote printing
- Controlling data flow through the `lpss`
- Priorities of printers and print requests

- Using plotters with the `lpss`
- Collecting and reporting statistics about data flow through the `lpss`

This chapter describes how to accomplish the tasks associated with these topics using SAM and SPP-UX commands. The SPP-UX commands method is provided for those who do not have access to SAM or choose not to use it.

If you are already familiar with the basic concepts of the `lpss` you may want to proceed directly to one of the tasks in the section of this chapter.

Note

If you are reading this chapter because you have a problem with the `lpss`, you should first refer to &sollp; in &solvepn;.

If you are working with the `lpss` for the first time or you want to review key concepts before performing a particular task, you should continue reading the material found in .

Terms used in this chapter

The following terms appear in discussions in this chapter. You can scan the list now and, if you need to, refer to it later.

destination

A print destination is a generic term used to describe a printer or printer class. Users can specify a print destination when they print something by using the `-d` option to the `lp` command. See **printer class**.

interface script

A shell script, located in the `/usr/spool/lp/interface` directory, that is the final stage of the `lpss`. Each printer that is configured in the `lpss` has an interface script that, under the control of the line printer scheduler, sends a print job to the printer.

intermixed listings

Multiple jobs printing on a printer simultaneously that result in printed pages with characters from different print jobs mixed together. The `lpss` is designed to prevent this.

line printer scheduler

The line printer scheduler is the heart of the `lpss`. It is the central program that must be running to ensure coordination of requests from users to printers.

lpss — The SPP-UX software subsystem responsible for controlling output to the printers on your system. Its primary responsibility is to prevent intermixed listings. It can also prioritize print jobs and start and stop output to printers.

local printer — A printer, configured into your `lpss`, that is physically connected to your computer. Local printers are not supported on CONVEX Exemplar systems. See **remote printer**.

logical printer — Each printer that is defined in your `lpss` is given a name that users will use to refer to it. You can create more than one definition (printer name) for any given printer. The logical printer name refers, not to the printer itself, but to one of the `lpss` definitions used to access the printer.

model script — When a printer is added to the `lpss`, an interface script must be created that can set up the communication to the printer and send data to it. Hewlett-Packard supplies models for these interface scripts in the `/usr/spool/lp/model` directory. These models are used by the lpadmin command to build the interface script for a printer as it is being defined.

cancel model script — The cancel model script, `/usr/spool/lp/cmodel/rcmodel`, is used to cancel a print request to a printer on a remote system.

status model script — The status model script, `/usr/spool/lp/smodel/rsmodel`, is used to return the status of the remote printer and print requests for the remote

network-based printer or plotter — A printer or plotter that is directly connected to the local area network.

print request — A term used to refer to a specific print job in the `lpss`.

print request identification number — The number the `lpss` uses to identify your print request. This identification number consists of the name of the printer or printer class followed by a sequence number.

print request priority — See **priority**.

printer class — A defined group of printers. A printer class can be used as a print destination instead of a printer name.

	The first available printer in the printer class will print the next job queued to that printer class.
printer fence priority	See **priority.**
printer name	When a printer is configured into the `lpss`, it is given a name that users can use to specify that printer as a print destination for their printout.
print queues	Also known as request directories, print queues are directories used by the `lpss` to hold print jobs for each print destination until they can be printed.
priority	A value associated with each printer and print request. Priorities are used to control which print requests can print on a given printer. Priorities can be adjusted and must have a range from 0 to 7.
remote print requests	A print request issued via the `lpss` on your system to be printed on a printer that is attached to a remote computer.
remote printer	A printer that is defined in your `lpss` but is physically connected to another computer (and accessed over a network).
remote spooling	The process used to allow printing to printers that are defined as part of your `lpss` but physically connected to another computer.
remote spooling daemon	A "behind the scenes" (background) program that runs on a remote computer. The remote spooling daemon receives print requests via a network and submits the print requests to its local `lpss` on the network user's behalf.
request directories	See **print queues.**
system default printer	When a user issues a print request, the user can specify a print destination. A system default printer can be defined so that, if a print destination is not specified, the `lpss` will use the system default printer.

LPspooler overview

You can think of the `lpss` as if it were a plumbing system. The data to be printed represents the "water" in this system. There are various request directories, sometimes referred to as printer queues, which serve as temporary holding tanks for print requests until they are sent to a printer to be printed. The flow of print requests is controlled at the request directory and printer level. The terms *accept* and *reject* refer to controlling the flow of print requests to the request directories while the terms *enable* and *disable* refer to controlling the flow of print requests to the printers. Accepting, rejecting, enabling, and disabling print requests control the data through the `lpss` as valves would control the flow of water in a real plumbing system. Shell scripts (called interface scripts) near the end of the data flow serve as pumps which "pump" an orderly flow of data to the printers.

The line printer scheduler controls the routing of print requests from the request directories to the printers. It functions as an automated flow controller to prevent **intermixed listings** and to provide efficient use of the printers on your system. Intermixed listings are multiple print requests printing on a printer simultaneously that result in printed pages with characters from different print requests mixed together.

You can add a printer to, or remove it from your system. If the "drain gets clogged" for one printer, you can re-route the print requests for that printer to another printer and you can "flush" unwanted print requests from the spooling system. You can also sent a print requests to another computer to printed. Sending print requests to another computer to be printed is called **remote spooling** and the other computer is referred to as a remote system. When you use remote spooling, a special shell script ("pump") is used to send the data to a remote system. A program on the remote system receives the data and directs it into the remote system's LP spooler.

The components of the LP spooler

The components of the `lpss` are:

- printer names
- printer classes
- print destinations
- system default printer
- printer interfaces
- printer models
- device files
- line printer scheduler
- local printer
- remote printer
- print request identification number

Printer names

When you configure a printer into the `lpss`, you assign it a name that you will use to refer to it when you later submit print requests. This name is referred to as the printer name. Printer names can contain up to 14 characters, which can be alpha-numeric or underscores. The name may or may not be the same as the device file name. Some correspondence between the printer name and device file name is suggested. The printer name is the name of the printer that shows up when you request the status of the printer queue with the lpstat command.

A hypothetical system "hypo1" has the following printers defined in its `lpss`. The printers have the following names:

- laser1
- laser2
- phred
- letterhead
- invoices
- check_printer

Printer classes

You can treat a group of printers as if they were one printer. A printer class is a name that you can use to refer to the group of printers. When submitting a print request you can specify a particular printer name or a printer class name. When submitting a print request to a printer class, the print requests will print on the first available printer in the group rather than on a specific printer. Printers that are members of a printer class can still be referenced individually. Creating a printer class is optional.

On the hypothetical system "hypo1," three of the printers are grouped into a printer class called "laser".

- printer class: laser
- laser1
- laser2
- phred
- Printer class names can contain up to 14 characters, which can be alpha-numeric or underscores.
- Printer class names and printer names on the same system cannot be the same name. Printer names and class names must all be unique.
- Printer classes cannot include remote printers.
- A printer class must contain at least one printer.
- A printer can only belong to one printer class at a time.
- To remove a printer from a printer class, you must remove the printer from the `lpss` and re-add without specifying a printer class.

Print destinations

Several of the commands for the `lpss` require you to specify a print destination. A destination is the name of a printer or printer class.

For our example system "hypo1", possible destinations are:

- printer class: laser
- laser1

- laser2
- phred
- invoices
- check_printer
- letterhead

System default printer (destination)

You can appoint one of the print destinations in your `lpss` to be the system default printer. It is not necessary to have a system default printer, but it is recommended. A system default printer receives any print requests that are not sent to a specific print destination. You can have only one system default printer.

In addition to, or instead of, a system default printer, you can assign each user a default printer to use. To do this, simply set the user's LPDEST shell environment variable to the name of the system default printer.

- If LPDEST is set and a user does not specify a different printer to use, the printer referenced by LPDEST will be used.
- If LPDEST is not set for a user, and the user does not specify a printer, the system default printer (if one is set) will be used.
- If neither LPDEST or the system default printer is set, a user must specify a printer (or printer class).

Printer interfaces

A printer interface, also known as an interface script, is the final stage of the `lpss`. It is the part of the `&lpss` that is responsible for sending data to a printer. Each printer that you have defined for use by the `lpss` has its own interface script (shell script) that resides in the `/usr/spool/lp/interface` directory. When printers are added to the `lpss`, an interface script is copied from `/usr/spool/lp/model` to `/usr/spool/lp/interface` and given the printer name.

If we were to list the directory `/usr/spool/lp/interface` on our hypothetical system "hypo1," it would contain the printer interface files `laser1`, `laser2`, `phred`, `letterhead`, `invoices`, and `check_printer`.

The entry for the class name laser would be located in the directory `/usr/spool/lp/class`; it would not be found in the interface directory.

The network-based printer script shipped from Roseville copies the system default interface script for the printer to `/usr/spool/lp/interface/model.orig` and renames the interface script the model or printer name. The script then replaces the system default script with a custom script in `/usr/spool/lp/interface`.

Printer models

There are printer interface script "models" you can choose from that have been created for you in the `/usr/spool/lp/model` directory. Many of them have names that match the model numbers of Hewlett Packard printers and plotters.

When you configure your printer into the `lpss`, you must specify which printer model interface script you want to use. The model will be automatically copied from the `/usr/spool/lp/model` directory into the `/usr/spool/lp/interface` directory and given the name that you specified as your printer name (see).

If you list the `/usr/spool/lp/model` directory, it should look similar to this:

A TABLE GOES HERE

If you have an HP printer, you will probably find a model script that matches its model number or name. Those interface model scripts that match your printers typically do not need to be changed. If you know how to do shell programming, you can customize printer interface model scripts to meet your specific printing needs (see <bookShells: User's Guide for information on shell programming).

Caution

The `update` program described in the &iu; manual can replace or remove model scripts in the process of updating your system. If you create your own printer interface scripts, keep the file names unique and keep a backup copy somewhere on the system.

If you do not have an HP printer, try using the dumb interface model. You might have to modify it to be able to use all of the features of your non-HP printer, but "dumb" should work for basic ASCII text printing. If the dumb printer interface model script does not work, contact your printer supplier for a UNIX line printer spooler interface script or try the script that most closely matches your non-HP printer type.

Device files

Device files are not part of the `lpss`; they are special files that define the necessary device driver and hardware address needed to communicate with a particular physical device (in this case a printer). The printer name referred to by the `lpss` and the name of the device file for a printer are not required to be the same, but a correspondence is recommended.

You can create printer device files using SAM or SPP-UX commands when you add a printer to the `lpss`. SAM creates a device file for you. If necessary, SAM can override the default device file naming convention. For information and specific instruction about how to make device files for your printers, see &sacpn; manual, &sacperiph; or the &pig; manual.

When you configure a printer into your `lpss`, you must supply the name of your printer's device file.

Line printer scheduler

The line printer scheduler is the heart of the `lpss`. It is the part of the `lpss` that prevents intermixed listings (output from more than one print request mixed together on a printed page) and controls flow of print requests to the printers. Its duties also include monitoring printer and print request priorities, monitoring/adjusting printer status, and logging `lpss` activities. The lpsched command starts the LP spooler. Because of the central role it plays, starting lpsched is referred to as "starting the

LP spooler", and stopping lpsched is often referred to as "stopping the LP spooler." You can use the `lpsched` command directly or through SAM (see and).

Local printer

A local printer is a printer that is physically connected to your system. Local printers are not supported on CONVEX Exemplar systems.

Remote printer

A remote printer is a printer that is not physically connected to your system, but can be accessed by your system through a local area network (LAN). To configure a remote printer into your local `lpss`, you must be able to access the remote system via a LAN.

Network-based printer/plotter

A network-based printer or plotter is connected directly to the local area network (LAN). A network-based printer or plotter is not physically connected to any system. This chapter provides instructions for setting up a network-based printer by means of SAM (see). If you do not prefer to use SAM, consult the instructions shipped with the printer or printer interface card product.

Print request identification number

When you submit a print request by means of the lp command, lp responds with a print request identification number consisting of the name of the printer (or printer class) followed by a number. Here are some examples of typical print request identification numbers:

```
laser-3456
phred-2152
letterhead-1547
```

Remote spooling

If you have several systems connected to a Local Area Network (LAN) and would like the systems to share the use of a printer, you can set up the `lpsss` of the systems that are not physically connected to the printer to automatically send their print requests (via the LAN) to the `lpss` of the system that does have the printer. The systems without printers act as though they were a user on the system with the printer, submitting print requests to that system's `lpss`. This is accomplished by a special program known as the Remote Spooling Daemon (rlpdaemon).

The rlpdaemon program runs in the background (on the system with the printer) monitoring the incoming LAN traffic for any remote print requests from other systems. When these requests arrive, the rlpdaemon program submits them to its local `lpss` on behalf of the remote user. In addition to remote print requests, the remote spooling daemon must also handle "cancel" and "status" requests from remote systems.

There are special "interface scripts" on the remote systems that issue cancel and status requests. These special interface scripts have a lot in common with printer interface scripts. They have a model directory that can hold sample versions of these scripts, and they have an interface directory where the scripts currently in use reside. The cancel and status models are copied into their respective interface directories automatically when adding a remote printer.

The directory `/usr/spool/lp/cmodel` contains a sample interface script, rcmodel, that sends a remote cancel command to the system with the printer. When you configure a remote printer into your `lpss`, the cancel model script is copied into the `/usr/spool/lp/cinterface` directory and is given the same name as the printer.

The directory `/usr/spool/lp/smodel` contains a sample of an interface script, called rsmodel, which sends a remote status command to the system with the printer. When you configure a remote printer into your `lpss`, the status model script is copied

into the /usr/spool/lp/sinterface directory and is given the same name as the printer. It is unlikely that you will need to customize the remote cancel and status model scripts. If you do customize these "remote control" scripts, you must copy them to a different file name to avoid destroying your changes when updating the system with the update utility.

Configuring a remote printer into your lpss requires that you inform your system of the following:

- The name of the system with the printer
- The interface script to use when it issues a remote cancel request
- The interface script to use when it issues a remote status request
- The name of the printer (as it is defined in the lpss of the remote system) where you want your printouts to be printed.

Priorities of printers and print requests

To control the order of printed requests, you can assign priority values to printers and to specific print requests. Assigning priorities is NOT required.

- Priority values must be in the range of 0 to 7.
- Priority 7 is the highest priority.
- A value assigned to each printer, known as a printer fence priority, determines the minimum priority that a print request must have in order to be able to print. A print request having a priority equal to or greater than the fence priority of its printer will print. SPP-UX assigns a printer fence priority value of zero (0) when you add a printer to the lpss. You can change printer fence priorities dynamically with SAM or the lpfence command.
- A value assigned to each print request, known as a print request priority, is associated with the destination printer. The print request priority for each printer can be determined when each printer is added to the line printer spooling system. If the printer print request priority is

changed after a print request has been put in the print queue, the print request's priority does not change.

- Print request priorities lower than the printer priority will not print. If a print request's priority is lower than its printer's priority, it will remain in the request directory ("printer queue") for that printer. It will remain there until its priority is raised or its printer's priority is lowered to allow it to print (or until the request is canceled).
- You cannot directly set a printer class priority. See for an example of a printer class. The class priority is the same as the highest priority of any printer in the class.
- If multiple print requests are waiting to be printed on a specific printer and all have priorities high enough to print:
  - The `lpss` will print next the print request with the highest priority.
  - If more than one print request has the highest priority, all print requests with that priority will print in the order they were received by the `lpss`.

Using plotters with the spooler

Because the `lpss` is nothing more than a data routing mechanism, it can be used with other output devices. Apart from printers, the devices most commonly used with the `lpss` are plotters. The following model scripts are supplied so that you can use your `lpss` with Hewlett Packard plotters:

LP Spooler Models for Plotters

A TABLE GOES HERE

Controlling data flow* through the spooler

There are three points in the `lpss` where you can control the flow of data:

1. You can start or stop the LP spooler. This has a global effect. If you stop the LP spooler, printing for all printers stops.
2. You can tell the `lpss` to accept or reject any new print requests for a printer. If you instruct the `lpss` to reject print requests for a printer class, users will be given a message telling them that the printer class that they requested is not accepting requests when they attempt to print something to that destination. Rejecting print requests should be used when a printer or a class of printers is being taken off the system for a extended period of time. Rejecting print requests is not recommended for making the printer unavailable for a short time. For example, rejecting print requests is not recommended for adding paper or changing the toner cartridge. A minor delay due to these short term services is usually acceptable.
3. You can tell the `lpss` to enable or disable a printer for printing. Print requests continue to be accepted for the disabled printer unless you have explicitly rejected print requests. Disable a printer should to make the printer temporarily unavailable for a short time, for example, disabling the printer to add paper or change toner. Do not disable a printer for a long time without also rejecting requests for that printer; otherwise users' print requests will keep accumulating in the print queue and they will not get any notice that their requests will not print. Once you reject print requests for a printer, a user submitting a print request to that printer will get a message stating that the printer is not accepting requests.

To print, a printer must be accepting and enabled.

Note

When you use SAM to "enable" or "disable" a printer, SAM performs both the accept/reject operation and the enable/disable operation listed above. If you wish to "disable" a printer but still accept requests for that printer (letting them accumulate in the request directory for the printer), you must use the SPP-UX commands method.

Logging and analyzing printer activity

Analyzing printer activity can help you determine if there are bottlenecks in your `lpss`. It can also help you determine/justify the need to add additional printers to your `lpss`. There are facilities to help you analyze the flow of data through your `lpss`.

There are two phases to analyzing `lpss` activity: a data collection phase and a data reporting phase. The data collection phase begins when the `lpss` starts. The -a option to the lpsched command turns on the data collection processes when you start the LP spooler (see). The data reporting phase can occur any time after the `lpss` has been started. The following statistics are calculated:

- average waiting time from when a print request is submitted to the start of printing
- standard deviation for waiting time
- average printing time from start to end of print request
- standard deviation of printing time
- average number of bytes (characters) printed per request
- standard deviation for number of bytes
- sum of bytes printed for all requests in Kbytes
- total number of requests since logging started

Initial spooler setup

Initial LP spooler setup consists of the following tasks:

1. Add at least one printer to the `lpss`.
2. Tell the `lpss` to accept print requests for this printer.
3. Tell the `lpss` to enable the printer for printing.
4. Turn on the LP spooler.

When you use SAM to add a printer, SAM:

- tells the `lpss` to accept print requests for the printer.
- enables the printer.
- starts the `lpss`.

If you are not using SAM, you must do these tasks yourself; refer to .

Spooler tasks

The two methods of controlling the `lpss` are:

1. The System Administration Manager (SAM)
2. SPP-UX commands

SAM allows you to control the `lpss` through its menu-selection and data-entry screens. By combining multiple "manual commands" into single tasks, SAM can save you time and keystrokes. SAM also eliminates the need to know command names and options for the `lpss`.

Although SPP-UX commands require you to learn more details than SAM does, you might need or prefer to use SPP-UX commands, for the following reasons:

- SPP-UX commands give you a greater degree of control over the `lpss`.
- SAM might not be configured into your system. You have to use SPP-UX commands to control the `lpss`.
- You might be more comfortable using SPP-UX commands.

- You may need to use the data collection facility. If you want to start data collection, you must use the lpsched -a command, not SAM, to start the `lpss`.

See Chapter 1 for additional information about using SAM.

To enter SAM; type:

- Ensure that you have superuser capabilities.
- Enter the `sam` command:

```
/usr/bin/sam
```

SAM will present you with the main window. For additional information about using SAM, activate the [[Help]] control button.

To exit SAM:

- If you located in the SAM main window or in a functional subarea menu, exit by activating the [[Exit]] control button.
- If you are within a functional area with an object list displayed, choose {{Exit}} from the "List" menu, and activate the [[Exit]] button from the main window.
- If you are within a dialog box, activate the [[Cancel]] button, chose {{Exit}} from the "List" menu, and then activate the [[Exit]] button from the SAM main window.
- You can also close the SAM main window with the window manager to exit SAM.

Additional task information

To perform system administration tasks on a remote system using SAM, run SAM, highlight {{Remote Administration}}, and activate the [[Open]] control button. For additional information refer to the SAM help system and the "Using SAM for Remote System Administration" section of Chapter 1 .

SAM is an optionally loadable part of SPP-UX. If you have not loaded SAM onto your system, you will not be able to use it. You can use the update program to add SAM to your system if you did not originally

load it and currently want to use it. For details about how to do this, see the &iu; manual.

Viewing printers and print request status using SAM

To view printers:

1. Run SAM; type:

   ```
   /usr/bin/sam
   ```

   See Chapter 1 for additional information about using SAM.

2. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.

3. Highlight {{Printers/Plotters }} and activate the [[Open]] control button.

To view print requests:

1. Run SAM; type:

   ```
   /usr/bin/sam
   ```

   See Chapter 1 for additional information about using SAM.

2. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.

3. Highlight {{Print Requests}} and activate the [[Open]] control button.

You can also view print requests by choosing {{Print Requests}} from the "List" menu within the "Printer/Plotter Manager" Window.

Additional task information

SAM provides an on line help system to assist you when you need additional information.

Activating the [[Help]] button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the "Help" menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the [[f1]] key gives you context-sensitive information for the object at the location of the cursor.

Note

The print request queue can change rapidly. To ensure that you are viewing the current data, choose {{Refresh List}} from the "Options" menu to view the current state of the print request queue.

The "Printer/Plotter Manager" object list displays the following information about the printers in the `lpss`:

- system default printer
- status of LP spooler (RUNNING or STOPPED)
- the printer name
- printer status (enabled, disabled, idle, busy)
- priority for each printer and printer class
- the printer type (local, remote, or network)
- the location of each printer (printer name and system for remote printers; no entry for network-based printers)

Adding a remote printer using SAM

1. Ensure that the remote system has the printer installed and configured into the remote system's line printer spooler system.
2. Gather the following information:
   - The name you are giving to this printer or plotter.
   - Whether or not you wish to make this device your system's default printer.
   - The name of the remote system to which the printer or plotter is attached.

- The name of the remote printer or plotter.
- The "cancel" model on the remote system (optional).
- The "status" model on the remote system (optional).
- Whether or not you wish to allow any user to cancel any printing request.
- Whether or not the remote printer is on a system using BSD (Berkeley Software Distribution) UNIX. Using BSD disables any lp -o*parm* options. BSD systems do not understand the -o option.

3. Run SAM; type:

```
/usr/bin/sam
```

See Chapter 1 for additional information about using SAM.

Note

During the process of adding and removing printers, SAM stops the `lpss` . Print requests being printed at the time the `lpss` stopped might not complete successfully. It is best to add a printer when there are no requests currently printing.

4. Highlight {{Peripheral Devices}} and activate the [[Open]] control button.
5. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.
6. Highlight {{Printers/Plotters}} and activate the [[Open]] control button.
7. Choose {{Add a remote printer/plotter >}} and the menu item associated with the printer interface type from the "Actions" menu.
8. Fill in the printer interface dialog box fields and turn on off check box values.

   Activating the [[Help]] button from a dialog or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

Pressing the [[f1]] key gives you context-sensitive information for the object field at the location of the cursor.

9. Activate the [[OK]] control button.

Additional task information

SAM provides an on line help system to assist you when you need additional information.

Activating the [[Help]] button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the "Help" menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the [[f1]] key gives you context-sensitive information for the object at the location of the cursor.

To configure a remote printer into your `lpss`, you must be able to access the system with the printer via a local area network (LAN).

Remote printers cannot be members of a printer class.

Adding a network-based printer using SAM

To add a network-based printer or plotter using SAM:

1. Ensure that the printer is connected to the network according to the installation instructions shipped with the network-based printer or the network interface card for the printer.
2. Gather the following information:
   - The name you are giving to this printer or plotter.
   - The printer node name.
   - The model or interface that the printer will use.
   - The link-level address of the network card installed in the printer.
   - The TCP-IP protocol printer requires an Internet Protocol (IP) address.
   - The priority for this printer.
   - The class to which the printer or plotter will be added (optional).

   In addition, decide whether or not you wish to make this device your system's default printer.

   Run SAM; type:

   ```
   /usr/bin/sam
   ```

   See Chapter 1 for additional information about using SAM.
3. Highlight {{Peripheral Devices}} and activate the [[Open]] control button.
4. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.
5. Highlight {{Printers/Plotters}} and activate the [[Open]] control button.
6. Choose {{Add a network-based printer}} then {{Add TCP-IP protocol printer...}} from the "Actions" menu.

7. Fill in the printer interface dialog box fields and turn on and off check box values.

   Activating the [[Help]] button from a dialog or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

   Pressing the [[f1]] key gives you context-sensitive information for the object field at the location of the cursor.

8. Activate the [[OK]] control button.

Additional task information

SAM provides an on line help system to assist you when you need additional information.

Activating the [[Help]] button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the "Help" menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the [[f1]] key gives you context-sensitive information for the object at the location of the cursor.

The software SAM needs to configure your network-based printer is shipped separately. Follow the instruction shipped with your printer to load the software.

Removing a printer using SAM

1. Run SAM; type:

   ```
   /usr/bin/sam
   ```

   See Chapter 1 for additional information about using SAM.
2. Highlight {{Peripheral Devices}} and activate the [[Open]] control button.
3. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.
4. Highlight {{Printers/Plotters}} and activate the [[Open]] control button.
5. Highlight the printer you want to remove in the object list.
6. Choose {{Remove a printer/plotter >}} from the "Actions" menu.

Additional task information

SAM provides an on line help system to assist you when you need additional information.

Activating the [[Help]] button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the "Help" menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the [[f1]] key gives you context-sensitive information for the object at the location of the cursor.

Note

During the process of adding and removing printers, SAM stops the `lpss`. Print requests being printed at the time the `lpss` is stopped might not complete successfully. It is best to stop the `lpss` when there are no requests currently printing.

SAM cancels all print requests in the request directory for the printer you are removing.

SAM does not remove the device file for the printer removed from the `lpss`.

SAM removes the printer device file from the system.

Starting and stopping the spooler using SAM

To start the LP spooler:

1. Run SAM; type:

   ```
   /usr/bin/sam
   ```

   See Chapter 1 for additional information about using SAM.

2. Highlight {{Peripheral Devices}} and activate the [[Open]] control button.
3. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.
4. Highlight {{Printers/Plotters}} and activate the [[Open]] control button.
5. Choose {{Start up printer spooler}} from the "Actions" menu.

To stop the LP spooler:

1. Run SAM; type:

   ```
   /usr/bin/sam
   ```

   See Chapter 1 for additional information about using SAM.

2. Highlight {{Peripheral Devices}} and activate the [[Open]] control button.
3. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.

4. Highlight {{Printers/Plotters}} and activate the [[Open]] control button.
5. Choose {{Shut down printer spooler}} from the "Actions" menu.

Additional task information

SAM provides an on line help system to assist you when you need additional information.

Activating the [[Help]] button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the "Help" menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the [[f1]] key gives you context-sensitive information for the object at the location of the cursor.

Note

Printing on all printers stops.

When SAM stops the `lpss` there is no guarantee that print requests being printed at the time will complete successfully. It is best to stop the `lpss` when there are no requests currently printing.

To turn on the data collection processes, refer to .

Determining the status of the spooler using SAM

To determine the status of the `lpss`:

1. Run SAM; type:

   ```
   /usr/bin/sam
   ```

   See Chapter 1 for additional information about using SAM.

2. Highlight {{Peripheral Devices}} and activate the [[Open]] control button.
3. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.
4. Highlight {{Printers/Plotters}} and activate the [[Open]] control button.

The status area of the object list will display the status of the scheduler as "Scheduler: RUNNING" or "Scheduler: STOPPED".

Additional task information

SAM provides an on line help system to assist you when you need additional information.

Activating the [[Help]] button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the "Help" menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the [[f1]] key gives you context-sensitive information for the object at the location of the cursor.

Disabling a printer using SAM

1. Run SAM; type:

   ```
   /usr/bin/sam
   ```

   See Chapter 1 for additional information about using SAM.
2. Highlight {{Peripheral Devices}} and activate the [[Open]] control button.
3. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.
4. Highlight {{Printers/Plotters}} and activate the [[Open]] control button.
5. Highlight the printer you want to disable in the object list.
6. Choose {{Disable printer}} from the "Actions" menu.

Additional task information

SAM provides an on line help system to assist you when you need additional information.

Activating the [[Help]] button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the [[f1]] key gives you context-sensitive information for the object at the location of the cursor.

It is best to disable printer when there are no requests currently printing.

Note

When you use SAM to "enable" or "disable" a printer, SAM performs both the accept/reject operation and the enable/disable operation. If you wish to "disable" a printer but still accept requests for that printer (letting them accumulate in the request directory for the printer), you must use the SPP-UX commands method to disable the printer (see).

Enabling a printer using SAM

To enable a printer using SAM:

1. Run SAM; type:

   ```
   /usr/bin/sam
   ```

   See Chapter 1 for additional information about using SAM.
2. Highlight {{Peripheral Devices}} and activate the [[Open]] control button.
3. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.
4. Highlight {{Printers/Plotters}} and activate the [[Open]] control button.
5. Highlight the printer you want to enable in the object list.
6. Choose {{Enable printer}} from the "Actions" menu.

Additional task information

SAM provides an on line help system to assist you when you need additional information.

Activating the [[Help]] button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the "Help" menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system

- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the [[f1]] key gives you context-sensitive information for the object at the location of the cursor.

Note

When you use SAM to "enable" or "disable" a printer, SAM performs both the accept/reject operation and the enable/disable operation. If you wish to "disable" a printer but still accept requests for that printer (letting them accumulate in the request directory for the printer), you must use the SPP-UX commands method to disable the printer (see).

Changing a printer fence priority using SAM

To change a printer priority using SAM:

1. Run SAM; type:

   ```
   /usr/bin/sam
   ```

   See Chapter 1 for additional information about using SAM.
2. Highlight {{Peripheral Devices}} and activate the [[Open]] control button.
3. Highlight {{Printers and Plotters}} and activate the [[Open]] control button.
4. Highlight {{Printers/Plotters}} and activate the [[Open]] control button.
5. Highlight the printer for which you want to change the priority.
6. Choose {{Modify fence priority}} from the `Actions` menu.
7. Choose the new priority value from the Printer priority menu button.
8. Activate the [[OK]] control button.

Additional task information

SAM provides an on line help system to assist you when you need additional information.

Activating the [[Help]] button from the SAM main window, a dialog box, or message box gives you information about the attributes and tasks you can perform from the currently displayed window.

From within a functional area, choosing an item from the `Help` menu gives you information about:

- the current functional area
- keyboard navigation within SAM
- using the SAM help system
- displaying the version of SAM you are currently running

From a dialog box (a window displaying fields to be filled in), pressing the [[f1]] key gives you context-sensitive information for the object at the location of the cursor.

See for additional information.

Setting up the spooler using SPP-UX commands

1. Add at least one printer to the `lpss` (see or).
2. Tell the `lpss` to accept print requests for this printer (see).
3. Tell the `lpss` to enable the printer for printing (see).
4. Start the LP spooler (see).

Determining if a device file exists for your printer using SPP-UX commands

1. Use the &pig; manual to help you determine what the minor numbers for your printer should be (based on the printer's interface and hardware address). The printer driver names, major number, and interface types are as follows:

   A TABLE GOES HERE

2. Use the ll command to list the directory /dev. Look through the entries for one that matches the major and minor numbers. The fifth column of information (immediately to the right of the group ownership) represents the major number for the corresponding device file. The sixth column (immediately to the left of the date) represents the minor number for the device file.

3. If you find one that matches, note its device file name (last column of information in the ll listing) for use with the SPP-UX commands to add a printer to your spooling system (later in this chapter).

   If you do *not* find one with major and minor numbers that match, you will need to create a device file for your printer. See the &pig; for the procedure on how to do this.

Additional task information

A device file is the mechanism that SPP-UX uses to determine which of the devices attached to your computer it should use for an I/O operation. The major number of the device file tells SPP-UX which drivers to use; the minor number tell SPP-UX the hardware address of the device. Device files are usually located in the /dev directory.

Adding a remote printer using SPP-UX commands

To add a remote printer using SPP-UX commands:

1. Ensure that you have superuser capabilities.
2. Stop the LP spooler with the `lpshut` command:

```
/usr/lib/lpshut
```

Note

It is best to stop the LP spooler when there are no requests currently printing.

3. Add the printer to the `lpss` using the lpadmin command:

```
/usr/lib/lpadmin -ppname -vdevfile
-mmodel [-d] [-gpriority]
[-ocmcmodel] [-osmsmodel]
[-ormremsys] [-orprpname] [-ob3]
[-orc]
```

where:

pname This is the name that you will use to send print requests to this printer. Printer names can be up to 14 characters in length, and the characters must either be alphanumeric (A-Z, a-z, 0-9) or an underscore (_).

devfile Since the printer is not physically connected to your local system, use the `/dev/null` device file.

model The remote model script is the `/usr/spool/lp/model/rmodel`. A copy of this file will be put in the `/usr/spool/interface` directory with the name you specified in *pname*.

`-d` specifies that you want this printer to be the system default printer.

priority You only need to use this option if you want your printer to have a

priority other than zero (optional).

cmodel The cancel model script `/usr/spool/lp/cmodel/rcmodel` is used to forward a `cancel` request over to the remote system's `lpss`. The `lpss` copies `/usr/spool/lp/cmodel/rcmodel` to the `/usr/spool/lp/cinterface` directory with the name you specified in *pname*.

smodel The status model script `/usr/spool/lp/smodel/rsmodel` is used to forward a `status` request over to the remote system's `lpss`. The `lpss` copies `/usr/spool/lp/smodel/rsmodel` to the `/usr/spool/lp/sinterface` directory with the name you specified in *pname*.

remsys The name of the remote system to which the printer is physically connected. You can get the remote system name by entering the command hostname (with no options) on the system with the printer. The name of the remote system must be available to the local system, either from a name server or in the `/etc/hosts` file on the local system (required). Need to reference a NETWORKING manual here.

rpname This is the printer name as it is defined on the remote system.

When using the lpadmin command, do not put any spaces between the options and their respective values. lpadmin will not interpret your input correctly if you do. For example:

TYPE THIS:

```
-pinvoices
```

NOT THIS:

```
-p invoices
```

4. Allow print requests to enter the request directory for the newly added remote printer with the accept command:

   ```
   /usr/lib/accept pname
   ```

   where:

   pname is the local name given to this remote printer.

5. Enable the newly added remote printer to process print requests with the enable command:

   ```
   /usr/bin/enable pname
   ```

   where:

   pname is the local name given to refer to this remote printer.

6. Start the line printer scheduler with the lpsched command:

   ```
   /usr/lib/lpsched
   ```

   See

Additional task information

A remote printer is a printer that is not physically connected to your system, but can be accessed by your system through a local area network (LAN). To configure a remote printer into your local `lpss`, you must be able to access the remote system via a LAN.

Remote printers cannot be members of a printer class.

Adding a printer to the `lpss` is not the same thing as adding a printer to your system. The first involves connecting the printer to your computer and configuring SPP-UX to communicate with the printer. The second involves configuring the software subsystem (known as the Line Printer Spooling System) that manages printer output.

Because the lpadmin is constructing and modifying files that are used by the line printer scheduler, it is important that the scheduler is stopped when you use the lpadmin command to add a new printer.

You only need to use the -ob3 option if your print request will be printed on or pass through a system that uses the Berkeley Software Distribution (BSD) style `lpss`. BSD systems use three-digit (rather than four-digit) print request-ID numbers (these are the numbers returned when you send something to print). The -ob3 option disables any lp -o*parm* options. BSD systems do not understand the -o option to the lp command.

Use the -orc if you want to restrict users to cancelling only their own print requests.

Examples

To determine the `lpss` status:

```
/usr/bin/lpstat -r scheduler is
stopped
```

To add a remote printer, referred to locally as letterhead, physically connected to the system hypo2

that uses the BSD style print request-ID numbers, and is known on the remote system as "memos":

```
/usr/lib/lpadmin -pletterhead
   -v/dev/null -mrmodel -ocmrcmodel
   -osmrsmodel -ormhypo2 -ob3
   -orpmemos
```

Note

Because there are so many options to the commands, these examples use a backslash (<esc) to represent a line continuation. When you type these commands, you can enter the backslash as shown or you can omit the backslash and type the entire command before pressing RETURN.

To add a remote printer, referred to locally as remote_drafts, physically connect to the system system13, known on the remote system as old_reliable, and requires a printer priority of 3:

```
/usr/lib/lpadmin -premote_drafts
     -v/dev/null -mrmodel -ocmrcmodel
     -osmrsmodel -ormsystem13 -g3
     -orpold_reliable
```

To allow print requests to enter the request directory for the newly added remote printers:

```
/usr/lib/accept letterhead
/usr/lib/accept remote_drafts
```

To enable the newly added remote printers:

```
/usr/bin/enable letterhead
/usr/bin/enable remote_drafts
```

To start the line printer scheduler, type:

```
/usr/lib/lpsched
```

Adding a network-based printer using SPP-UX commands

To add a network-based printer or plotter using SPP-UX commands, follow the instructions shipped with the network-based printer or the network interface card for the printer.

Additional task information

The software needed to configure your network-based printer is shipped separately. Follow the instruction shipped with your printer to load the software and configure the printer.

Creating a printer class using SPP-UX commands

To create a class of printers, use the -c option to the lpadmin command when you add a printer to the `lpss` or after you have added several printers to the `lpss`. A printer class must contain at least one printer. See for instructions on creating a printer class as you add a printer to the `lpss`.

To create a printer class after several printers have been added to the `lpss`:

1. Ensure that you have superuser capabilities.
2. Stop the `lpss` (see).

Note

It is best to stop the LP spooler when there are no requests currently printing.

3. Create the printer class by entering the lpadmin command, specifying the -c option, for every printer you wish to add to a class of printers. There is an example at the end of this section.
4. Start the `lpss` (see).
5. Allow print requests to enter the request directory for the newly added printer class with the accept command:

```
/usr/lib/accept pname
```

where:

pname is the name given to this printer class.

Additional task information

Printer classes cannot include remote printers.

A printer can only belong to one printer class at a time. To remove a printer from a printer class, remove the printer from the `lpss` and re-add the printer omitting the -c option of the lpadmin command.

It is not necessary to specify the model and device file options because the printers have already been defined for the `lpss`.

Printer class names can be up to 14 characters in length, and the characters must either be alphanumeric (A-Z, a-z, 0-9) or an underscore (_). Note that class names and printer names on the same system cannot be the same name. Class and printer names must be unique. A printer can only belong to one printer class at a time.

Examples

To create a laser class of printers consisting of printers *laser1*, *laser2*, and *phred*:

```
/usr/lib/lpadmin -plaser1 -claser
/usr/lib/lpadmin -plaser2 -claser
/usr/lib/lpadmin -pphred -claser
```

To remove a printer from a printer class, remove the printer from the `lpss` and re-add the printer without the printer class (see and).

Removing a printer or printer class using SPP-UX commands

To remove a printer or printer class using SPP-UX commands:

1. Ensure that you have superuser capabilities.
2. Deny any further print requests for the printer with the reject command:

   ```
   /usr/lib/reject [-r "message"]
     name [name]
   ```

   where:

   message is a message to be displayed when users obtain status information about the printer(s) and/or printer classes. *name* is the name of the printer or printer class.

3. (If you are removing a printer class, skip this step and continue with step 4.)

   Disable the printer to be removed with the disable command:

   ```
   /usr/bin/disable [-r "message"]
     [-c] pname [pname]
   ```

   where:

   message is a message to be displayed when users obtain status information about the printer(s). *pname* is the name of the printer to be disabled.

Note

When you disable a printer, any print requests waiting to be printed for that printer will remain in the printer's request directory. If you wish to cancel all print requests for a printer at the time you disable it, use the `-c` option with the `disable` command:

```
/usr/bin/disable -c letterhead
```

4. Stop the `lpss`:

   ```
   /usr/lib/lpshut
   ```

   Before you stop the line printer scheduler (spooling system), beware of the following:

   - All printing will stop until you restart the scheduler.
   - Any print requests that are currently printing will be completely reprinted when you restart the scheduler. This includes the print requests that were printing page 9,999 of a 10,000 page printout.

5. To preserve the print requests in the request directory, move all print requests in the request directory for the printer or printer class to another printer or printer class request directory (see).

   ```
   /usr/lib/lpadmin -x name
   ```

   where:

   name is the name of the printer or printer class to be removed.

6. If you have just removed your only printer, omit this step.

   Start the `lpss` (see):

   ```
   /usr/lib/lpsched
   ```

Additional task information

Because `lpadmin` is deleting and modifying files that are being examined by the line printer scheduler, it is important that the scheduler is stopped when you use the lpadmin command to remove the printer from the `lpss`.

When you remove a printer class, the printers in it are not removed. You can still use them as individual printers. If the only printer in a printer class is removed, the printer class is removed also.

Examples

To remove the laser1 printer:

```
/usr/lib/lpadmin -xlaser1
```

To remove the laser printer class:

```
/usr/lib/lpadmin -xlaser
```


Accepting and rejecting print requests for a printer using SPP-UX commands

To accept print requests for a printer or printer class, use the accept command:

```
/usr/lib/accept name [name]
```

where:

name is the name of the printer class whose request directory is to be enabled to receive print requests.

You can issue individual commands for each printer class or you can combine the printer classes in one command.

To reject print requests for a printer or printer class, use the reject command:

```
/usr/lib/reject [-r"message"] name
   [-r"message"] [name]
```

where:

message is a message to be displayed when users obtain status information about the printer or printer class. *name* is the name of the printer or printer class whose request directory is being prohibited from receiving print requests.

You can issue individual commands for each printer class or you can combine the printer classes in one command separated by spaces. If you combine them, you can also specify different reasons for rejecting printer requests for different printers (and printer classes).

Additional task information

Even if all printers that are members of a class are accepting requests, the class can still reject requests. If that were the case, users would need to specify a specific printer, not the class, in later print requests.

If all printers in a class are rejecting requests, but the class itself is accepting requests, the print requests will remain in the request directory until at least one of the printers in the class begins to process print requests.

If you do not specify a reason, the status requests will get the response

```
Printer  is NOT ACCEPTING requests:
Reason is unknown.
```

Examples

To accept print requests for the laser1, laser2, phred, invoices, check_printer printers and the laser printer class:

```
/usr/lib/accept laser1
/usr/lib/accept phred
/usr/lib/accept invoices laser2
check_printer laser
```

To reject print requests for the laser1, laser2, phred, invoices, check_printer printers and the laser printer class:

```
/usr/lib/reject -r"Printer on loan to
   seismology lab." laser1
/usr/lib/reject -r"Printers being
   serviced" laser1 check_printer
/usr/lib/reject -r"Invoice forms on
   order" invoices -r "printers are
   being serviced" laser1 laser2
   phred laser
```

Note

A backslash (<esc) is used to represent a line continuation. When you type these commands, you can enter the backslash as shown or you can omit the backslash and type the entire command before pressing RETURN.

Enabling or disabling a printer using SPP-UX commands

To enable a printer to process print requests, use the enable command:

```
/usr/bin/enable pname [pname]
```

where:

pname is the name of the printer to be enabled to process print requests.

You can issue individual commands for each printer or you can combine the printers in one command separated by spaces.

To disable a printer to process print requests, use the disable command:

```
/usr/bin/disable [-r"message"] pname
   [-r"message"] [pname]
```

where:

message is a message to be displayed when users obtain status information about the printer(s). *pname* is the name of the printer to be disabled to process print requests.

You can issue individual commands for each printer class or you can combine the printer classes in one command separated by spaces. If you combine them, you can also specify different reasons for disabling printer requests for different printers (and printer classes).

Additional task information

Note

When you disable a printer, any print requests waiting to be printed for that printer will remain in the printer's request directory. When the printer is enabled again, the print requests will print. Any print request that are printing at the time the disable command is issued will be completely reprinted when the printer is enabled. If you wish to cancel all print requests for a printer at the time you disable it, use the `-c` option with the `disable` command:

```
/usr/bin/disable -c letterhead
```

Examples

To enable the check_printer, laser1, laser2, and phred printers:

```
/usr/bin/enable check_printer
/usr/bin/enable laser1 laser2 phred
```

To disable the check_printer, invoices, phred, letterhead, and laser printers:

```
/usr/bin/disable check_printer
/usr/bin/disable invoices phred
    letterhead
/usr/bin/disable -r "printer disabled
    to change paper" laser1
```

Setting a printer fence priority using SPP-UX commands

To set or change a printer priority:

1. Ensure that you have superuser capabilities.
2. Stop the `lpss` (see).

Note

It is best to stop the LP spooler when there are no requests currently printing.

3. Use the lpfence command to set priority for a particular printer:

   `/usr/lib/lpfence` *pname priority*

   where:

   pname is the printer name. *priority* is the minimum required priority a print request must have in order to be printed on printer *pname*. Fence value range is 0 (lowest) to 7 (highest).

4. Restart the `lpss` (see):

   `/usr/lib/lpsched`

Additional task information

When a printer is added to the `lpss`, the default priority is set to 0 (see).

Starting and stopping the spooler using SPP-UX commands

To start the `lpss`, use the lpsched command:

```
/usr/lib/lpsched
```

To stop the `lpss`, use the lpshut command:

```
/usr/lib/lpshut
```

Additional task information

Before you stop the line printer scheduler (spooling system), beware of the following:

- All printing will stop until you restart the scheduler.
- Any print requests that are currently printing will be completely reprinted when you restart the scheduler. This includes the print requests that were printing page 9,999 of a 10,000 page printout.

In order to report statistics about data flow through your `lpss`, you must tell the `lpss` that you want it to keep track of these statistics by specifying the -a option when starting the `lpss` with the lpsched command. The -a option tells the `lpss` to log statistical information about its activities to the file `/usr/spool/lp/lpana.log`, a file which will be used by the lpana command to report the statistics.

Examples

To find out `lpss` status:

```
/usr/bin/lpstat -r "scheduler is
  stopped"
```

To collect statistics about the data flow through the `lpss`, start the `lpss` with the `-a` option:

```
/usr/lib/lpsched -a
```

Canceling print requests using SPP-UX commands

To cancel print requests, use the cancel command:

```
/usr/bin/cancel req-ID [printer]
```

where:

req-ID is the print request identification number.
printer is the printer name.

You can issue individual commands for each print request or you can combine the print requests in one command separated by spaces.

You do not need superuser capabilities to use the cancel command.

Additional task information

To list print request identification numbers, use the `lpstat` command (see).

The cancel command has several useful options that allow you to do things such as cancel all print requests that you have submitted or cancel all requests associated with a particular printer or printer class. Here are a few helpful cancel options and their descriptions:

`-a`	Remove all requests a user owns on the specified printer. The owner is determined by the user's login name and host name on the machine where the `lp` command was invoked
`-e`	Empty the spool queue of all requests for the specified printer. Only users with superuser capabilities can use the -e option
`-i`	Cancel only local requests.

-u *user* Remove any requests queued belonging to user. Multiple `-u` options are allowed. Only users with superuser capabilities can use the `-u` option.

Examples

```
cancel laser-3456 cancel phred-2152
cancel letterhead-1547
```

or

```
cancel laser-3456 phred-2152
   letterhead-1547
```


Moving all requests using SPP-UX commands

To move all print requests to another request directory using SPP-UX commands:

1. Ensure your have superuser capabilities.
2. Prohibit any further requests from entering the request directory with the reject command:

   `/usr/lib/reject` *name* `[`*name*`]`

   where:

   name is the name of the printer or printer class request directory to be enabled to receive print requests.

   You can issue individual commands for each printer class or you can combine the printer classes in one command separated by spaces. If you combine them, you can also specify different reasons for rejecting printer requests for different printers (and printer classes).
3. Disable the printer with the disable command:

   `/usr/bin/disable [-r"`*message*`"]`
   pname `[-r"`*message*`"] [`*pname* `]`

   where:

   message is a message to be displayed when users obtain status information about the

printer(s). *pname* is the name of the printer to be disabled to process print requests.

You can issue individual commands for each printer class or you can combine the printer classes in one command separated by spaces. If you combine them, you can also specify different reasons for disabling printer requests for different printers (and printer classes).

4. Stop the `lpss` with the lpshut command:

```
/usr/lib/lpshut
```

5. Relocate all of the print requests in the request directory to another request directory with the lpmove command:

```
/usr/lib/lpmove source dest
```

where:

source is the printer or printer class request directory that you want to move to the *dest* request directory. *dest* is the printer or printer class request directory to receive the print requests from the *source* request directory.

6. Restart the line printer scheduler with the lpsched command:

```
/usr/lib/lpsched
```

7. If the *source* printer or printer class is to be made available to receive print requests:

   - Re-enable the printer(s) to process print requests with the enable command:

```
/usr/bin/enable pname [pname]
```

where:

pname is the name of the printer to be enabled to process print requests.

You can issue individual commands for each printer or you can combine the printers in one command separated by spaces.

- Re-enable the printer or printer class request directory to accept print requests with the accept command:

  ```
  /usr/lib/accept name [name]
  ```

  where:

  name is the name of the printer or printer class request directory to be enabled to receive print requests.

  You can issue individual commands for each printer class or you can combine the printer classes in one command.

Examples

To move all print requests from laser1 request directory to phred request directory:

```
/usr/lib/reject laser1
/usr/bin/disable laser1
/usr/lib/lpshut
/usr/lib/lpmove laser1 phred
/user/lib/sched
/usr/bin/enable laser1
/usr/lib/accept laser1
```

Moving selected print requests using SPP-UX commands

To move selected print requests to another request directory using SPP-UX commands:

1. Ensure that the `lpss` is running.
2. Move selected print requests using the lpalt command:

   ```
   /usr/bin/lpalt source -ddest
   ```

   where:

   source is the identification number of the print request to be moved. *dest* is the printer or printer class request directory to receive the print request specified by *source*.

Additional task information

The `lpalt` command cannot be used to alter a print request that is currently printing.

The `lpalt` command will alter a print request from a remote printer only if the print request is owned by the user who is issuing the `lpalt` command and, again, this alteration will only take place if the print request is not currently printing.

Examples

To move print request *laser-6610* to `phred` request directory:

```
lpalt laser-6610 -dphred new request
id is phred-6613
```

Viewing the status of printers and print requests using SPP-UX commands

To view the status of printers and print requests, use the lpstat command:

```
/usr/bin/lpstat [-t]
```

If no options are given, lpstat displays the status of all requests made by the user. The -t option lists the following additional information:

- status of the `lpss`.
- system default printer.
- list of class names and their members.
- list of printers and associated device files.
- status of each print request directory (accepting or rejecting). If a reason was specified when the requests were rejected the reason is displayed.
- status of each printer (enabled or disabled). If a reason was specified when the printer was disabled, the reason is displayed.
- priority for each printer.

- list of print requests for each printer that includes the following attributes for each print request:
  - print request identification number
  - name of user that submitted the print request
  - priority
  - date and time submitted
  - file name
  - size

Additional task information

The `-t` option of the `lpstat` command is very detailed. For information on other options of this command, refer to lpstat(1) man page in the &hpuxrefpn;.

Examples

To display a summary status of the lpss:

```
lpstat -t

scheduler is running system default
destination: laser members of class
laser: laser1 laser2 phred device
for letterhead: /dev/null device for
check_printer: /dev/null device for
laser1: /dev/lj1 remote to: shasta
on mountian device for laser2:
/dev/lj2 remote to: hood on mountian
device for phred: /dev/lj3 device
for invoices: /dev/invoices laser1
accepting requests since Apr 18
14:46 laser2 accepting requests
since May 13 14:08 phred accepting
requests since Apr 18 14:46 laser
accepting requests since Apr 18
14:46 letterhead accepting requests
since Apr 18 14:46 invoices
accepting requests since Apr 18
14:56 check_printer accepting
requests since May 3 14:57 printer
laser1 now printing laser1-1807.
enabled since Apr 23 13:47 fence
priority : 0 printer laser2 now
printing laser2-1809. enabled since
Apr 23 13:47 fence priority : 0
printer phred is idle. enabled since
Apr 18 14:46 fence priority : 3
printer letterhead now printing
letterhead-1810. enabled since Apr
23 13:47 fence priority : 4 printer
invoices is idle. enabled since Apr
19 10:24 fence priority : 0 printer
check_printer is idle. enabled since
Apr 18 14:56 fence priority : 0
laser1-1808 susanl priority 0 Jun 14
10:05 on laser1 disktab 5808 bytes
laser1-1809 susanl priority 0 Jun 14
10:05 report1 17301 bytes
laser2-1810 kimj priority 0 Jun 14
10:07 on laser2 memokmj 947 bytes
letterhead-1811 johnc priority 4 Jun
14 10:09 on letterhead salaries 2999
bytes
```

Changing the priority of print requests using SPP-UX commands

To change the priority of a print request, use the lpalt command:

```
/usr/bin/lpalt preq-ID -pnew_priority
```

where:

preq-ID is the print request identification number for the print request targeted for a new priority.
new_priority is the new priority. Valid values are 0 to 7.

Additional task information

There are two primary reasons for changing a print request priority:

1. To move the print request ahead of other requests within the request directory.

   For example, you can change the priority of your print request to be higher than that of the large print request that is ahead of yours. When the line printer scheduler selects the next print request to send to the printer, it will take the one with the highest priority (which is now yours because you changed the priority).

Note

Once a print request is printing, it will not yield to a print request of higher priority. In this case, you can move your print request to another printer if possible. See for details.

2. To match or exceed the printer's priority, enabling the print request to be processed (see).

Unless you tell it otherwise, the `lp` command (used to print things) will assign your print request a priority equal to that of its printer's printer priority setting. If your print request is assigned to a printer class, the highest printer priority setting among all the printers in the class will be used.

Examples

To find the following print request information:

- The print request-ID for the print request you want to change
- The current priority of the print request
- The priorities of the other print requests on the same printer
- The priority of the printer

Use the lpstat command:

```
lpstat

phred-1827 stevenm priority 0 Jun 14
   10:05 on phred
   proglisting 1708 bytes
phred-1828 paulv priority 2 Jun 14
   10:05 LONGproglist 6900714 bytes
phred-1829 chrisn priority 1 Jun 14
   10:05 urgentmemo 311 bytes
```

Displaying statistics about printer activity using SPP-UX commands

Note

Prior to displaying statistics about printer activity, the `lpss` must have been started with the `/usr/lib/lpsched -a` command to create a log of activity in the `/usr/spool/lp/lpana.log` file.

To display statistics about printer activity, use the `lpana` command:

```
/usr/lib/lpana [-ddest]
```

where:

dest defines the printer or printer class for which statistics are displayed. By default, `lpana` will report statistics for all printers and printer classes (optional).

Additional task information

Interpreting lpana's Output

A HUGE TABLE GOES HERE

Examples

To display statistics for all printers:

```
/usr/lib/lpana
```

```
performance analysis is done from Jun.22 '90 14:02 through Jun.27 '90
15:29
---printers  ----wait----  ---print---  ---bytes---     -sum-  num_of
/classes--   AV     SD     AV     SD     AV     SD     KB     requests
letterhead   0'00   0      0'49   2      59565  0      116    2
phred        0'00   0      0'45   22     14202  0      166    12
check_printer 0'09 31      0'51   73     12378  0      302    25
laser1       0'02   5      0'04   1      36686  0      2400   67
laser2       3'45   0      1'45   0      783    0      1      1
```

To display statistics for the laser printer class:

```
/usr/lib/lpana -d laser
```

```
performance analysis is done from Jun.22 '90 14:02 through Jun.27 '90
15:29
---printers  ----wait----  ---print---  ---bytes---     -sum-  num_of
/classes--   AV     SD     AV     SD     AV     SD     KB     requests
laser1       0'02   5      0'04   1      36686  0      2400   67
laser2       3'45   0      1'45   0      783    0      1      1
```

System accounting 8

SPP-UX allows concurrent sharing of computer resources among users: several users can be logged in, all sharing disk space, memory, and the CPU. SPP-UX System Accounting provides the means to:

- Monitor disk space usage for individual users
- Record connect session data (logins/logouts)
- Collect resource utilization data (such as memory usage and execution times) for individual processes
- Charge fees to specific users
- Generate summary files and reports that can be used to analyze system performance and bill users for resource consumption

SPP-UX System Accounting allows you to accomplish accounting tasks through a number of versatile commands. This chapter illustrates the use of these commands and contains the following sections:

Installation and daily usage

The purpose of this section is to show you:

- What you must do to get System Accounting running on your system
- How System Accounting automatically creates daily and monthly accounting data and reports

After reading this section, you should be able to install System Accounting on your system. Once properly installed, System Accounting will automatically generate daily and monthly accounting data and reports.

How to install System Accounting

Not all users require accounting services on their systems. For this reason, System Accounting is provided as an option: if you want to use System Accounting, you must install it yourself. The installation procedure is covered here.

There are three steps in the installation process:

1. Update `/etc/rc`
2. Create crontab entries
3. Set PATH for accounting commands

Each of these steps must be carried out to insure that System Accounting automatically creates daily and monthly accounting information. Detailed descriptions of each step follow.

Update /etc/rc

The system initialization shell script rc must be updated to automatically start System Accounting when the system is started in multiuser mode. This requires adding the following entry in the localrc section of /etc/rc:

```
/bin/su - adm -c /usr/lib/acct/startup
```

Create crontab entries

To automate the daily and monthly creation of accounting data, you should create a crontab file that cron can use to automatically run certain accounting commands. This process entails the following steps:

1. Log in to System Accounting as the user adm.
2. Use an editor to create the crontab file containing the accounting commands that are to be run automatically by cron. (The actual entries to make in this file are shown after these steps.)
3. Execute the crontab command, specifying the file created in step 2 as input. This step insures that the crontab file created in step 2 will be scanned by cron every minute. After invoking this command, the step 2 file will be stored in the file:

```
/usr/spool/cron/crontabs/adm
```

4. At this point, you are finished creating crontab entries. If you ever want to change the entries, simply re-edit the file created in step 2 and use the crontab command again. See the crontab(1) man page for more information.

The following entries, accompanied by a description of each, should be made in the crontab file created in above:

```
0 4* 1-6 /usr/lib/acct/runacct 2> /usr/adm/acct/nite/fd2log
```

`runacct`, the main accounting shell script, should be executed daily (during non-prime hours) to generate daily accounting reports. The above entry executes runacct at 4:00am every Monday through Saturday. Error messages will be redirected to the file `/usr/adm/acct/nite/fd2log`, if any errors occur while runacct executes.

```
0 2* 4 /usr/lib/acct/dodisk
```

`dodisk` creates total accounting records that summarize disk space usage for individual users. This entry runs `dodisk` at 2:00am every Thursday morning.

```
5 * * * * /usr/lib/acct/ckpacct
```

To insure that the process accounting file, pacct, doesn't get too large, the command `ckpacct` should be executed hourly. This entry invokes `ckpacct` at five minutes into every hour.

```
15 5 1* /usr/lib/acct/monacct
```

The monthly merging of accounting data is facilitated through the `monacct` command. This entry allows `monacct` to generate a monthly total report and total accounting file. monacct will be executed at 5:15am on the first day of every month.

The dates and times shown in the crontab entries above are only suggestions; you can tailor crontab entries to suit your needs. However, if you use different entries than those shown here, be sure that monacct is run at such a time as to allow runacct sufficient time to finish.

Set PATH for accounting commands

Finally, you should set the PATH shell variable in /usr/adm/.profile so that System Accounting knows where to look for commands. Path should be set as follows:

```
PATH=/usr/lib/acct:/bin:/usr/bin:/etc:/usr/adm
```

Summary of daily operation

The daily operation of System Accounting is summarized by the following steps:

1. When SPP-UX is switched into multiuser mode, the system initialization shell script `rc` executes the accounting command `startup`. The purpose of `startup` is to start System Accounting, and it performs the following functions:
   - Calls `acctwtmp` to add a boot record to wtmp. This record is marked by storing "acctg on" in the device name field of the `wtmp` record.
   - Turns process accounting on via turnacct on. turnacct on executes accton with the filename argument `/usr/adm/pacct`.
   - Removes work files left in the sum directory by `runacct`.
2. A report of the previous day's accounting information can be created by running `prdaily`. Obviously, this step is omitted the first day that System Accounting is installed, because the previous day's accounting information doesn't exist yet. However, after `runacct` has been executed, `prdaily` will generate valid reports.
3. The `ckpacct` command is executed every hour via cron to insure that the process accounting file `pacct` doesn't become too large. If pacct grows past a set maximum number of blocks, `turnacct switch` is invoked, which creates a new pacct file. (Other conditions may also limit the size of the process accounting file or turn process accounting off; for more details, see the discussion of `ckpacct` in the "Process

accounting" section of this chapter.) The advantage of having several smaller `pacct` files is that `runacct` can be restarted faster if a failure occurs while processing these records.

4. The `chargefee` program can be used to charge fees to users. It adds records to the file fee. These records are processed during the next execution of runacct and merged in with total accounting records.
5. `runacct` is executed via `cron` each night. It processes the active fee file and the process, connect session, and disk total accounting files. It produces command and resource-usage summaries by login name.
6. When the system is turned off using `shutdown`, the `shutacct` command is executed. The purpose of `shutacct` is to stop System Accounting, and it performs the following functions:
   - Writes a termination record to `wtmp` via the command `acctwtmp`. This record is marked by having "acctg off" in the device name field.
   - Turns process accounting off by calling `turnacct off`.

Overview of System Accounting

In this section, the intrinsics of System Accounting are examined. Key terms are defined, commands are introduced, system data flow is described, and finally, you are shown the login and directory structure of System Accounting.

Definitions

The following terms are specific to System Accounting.

Prime/non-prime connect time

Prime time is the time during the day when the computer system is most heavily used; for example, from 9:00am to 5:00pm. Non-prime time is the remaining time during the day when the system is

less heavily used; from 5:00pm to 9:00am in this example.

When reporting computer time usage, System Accounting distinguishes between prime and non-prime time usage. You can specify prime and non-prime time on your system by editing the file `/usr/lib/acct/holidays`. For details on the `holidays` file, see the section "Updating the holidays file" in this chapter.

Prime time is in effect only on weekdays (Monday through Friday); non-prime time is in effect during the weekends (Saturdays and Sundays) and on any holidays specified in the `holidays` file.

Process accounting records

Once System Accounting is installed and turned on, the following occurs: whenever a process terminates, the kernel writes a process accounting record for the terminating process into the current process accounting file, `/usr/adm/pacct` by default. (You can specify that a file other than `pacct` be used as the process accounting file, if desired.)

A process accounting record contains resource-usage data for a single process; it summarizes how much of the various resources the process used during its lifetime. Examples of information contained in process accounting records are:

- the user ID of the process's owner
- the name of the command that spawned the process
- the amount of time it took the process to execute

For greater detail on the contents and format of process accounting records, see the `acct(4)` man page.

Total accounting records

These records, created by various accounting commands, contain summary accounting information for individual users. These records provide the basic information for many reports generated by System Accounting. Some examples of information contained in these records are:

- the ID and user name of the user for whom the total accounting record was created
- the total number of processes that the user has spawned during the accounting period for which the total accounting record was created
- fees for special services rendered to this user

The exact contents and format of total accounting records can be found in the acct(4) man page. In addition, commands covered in later sections of this chapter show how these records are created and used by System Accounting.

Introduction to commands

System Accounting provides many versatile commands to accomplish numerous, varied tasks. There are commands that create data, commands that display data, commands that remove data, commands that merge data, and commands that summarize and report data. In addition, the output of one command may become the input to other commands.

System Accounting commands can be logically categorized into six basic command groups:

- installation
- disk usage accounting
- connect session accounting
- process accounting
- charging fees
- summarizing and reporting accounting information

Descriptions of these command groups, along with a brief synopsis of each command, follow.

Installation

These commands insure that System Accounting is properly installed. They are used to turn accounting on when SPP-UX is powered up and turn accounting off when the system is shut down. They may also do some file cleanups. Two such commands exist:

`startup`	Starts accounting when SPP-UX is switched to multiuser mode. startup is invoked from `/etc/rc`.
`shutacct`	Turns off accounting when SPP-UX is turned off via the `/etc/shutdown` shell.

Disk space usage accounting

In general, these commands produce disk usage accounting information: they show disk space usage (in blocks) for individual users. They also produce total accounting records. There are four commands:

`acctdusg` and `diskusg`	Both commands show how many blocks of disk space users are consuming. They differ in command options, and the manner in which they produce the information; `acctdusg` takes its input from a list of path names created by find, and `diskusg` looks at the inodes of the file system to create its output.
`acctdisk`	Produces total accounting records. Its input is supplied (either directly or indirectly) from `acctdusg` or `diskusg`.
`dodisk`	Produces total accounting records by using the `diskusg` and `acctdisk` commands. dodisk is normally invoked by `cron`.

Connect session accounting

Independently of System Accounting, the programs login and init record connect sessions by writing records into `/etc/wtmp`. System Accounting commands can display or fix this file, and can produce total accounting records for this file. There are six commands:

`acctwtmp`	Writes records to wtmp.
`fwtmp`	Displays the information contained in wtmp.
`wtmpfix`	Normalizes connect session records that span date changes (see the date(1) man page). Also validates login names in connect session records.
`acctcon1`	Summarizes wtmp in ASCII readable format, producing one line per connect session.
`acctcon2`	Takes input of the format produced by acctcon1 and produces total accounting records as output.
`prctmp`	Displays the session record file, normally called:

/usr/adm/acct/nite/ctmp

Process accounting

When process accounting is turned on, the kernel writes a process accounting record to pacct whenever a process terminates. A number of accounting commands exist that summarize and report this accounting information. In addition, certain commands turn process accounting on or off and insure that pacct doesn't become too large. The process accounting commands are:

`accton`

Turns process accounting on or off, depending on whether or not a filename argument is supplied with the command. If no filename is given, then process accounting is turned off; the kernel stops writing process accounting records to pacct. If a filename is specified, then the kernel starts writing process accounting records to the specified filename.

`accton` uses the system call `acct` to turn process accounting on or off. Only the superuser can execute `accton`.

`ckpacct`

Checks the size of the process accounting file `pacct`. If `pacct` becomes too large, then a new pacct file is created via turnacct switch. If disk space becomes critically short, then process accounting is turned off until sufficient space is available. This command is normally invoked by `cron`.

`turnacct on || off || switch`

Performs one of three functions, depending on which argument (on, off, or switch) is specified. turnacct on turns process accounting on by calling accton with the default filename argument `/usr/adm/pacct`; `turnacct` off turns process accounting off by calling accton with no filename argument; turnacct switch renames the current `pacct` file (so that it is no longer the current process accounting file) and creates a new, empty `pacct` file.

`acctcom`

Displays process accounting records contained in `pacct` (or any specified file).

`acctcms`

Takes `pacct` as input, and produces summary accounting information by command, as opposed to by process.

`acctprc1` Produces readable process accounting information, mainly for input into `acctprc2`.

`acctprc2` Takes input of the form produced by `acctprc1` and produces total accounting records.

Charging fees

Occasionally, you may want to charge a user for something. For example, you might charge fees to users for fixing any damaged files that they have. The chargefee command allows you to charge fees to specific users.

Summarizing and reporting accounting information

This group of commands summarizes and reports the data created through the command groups described above. These are the commands that are probably used most frequently; they represent the highest level of accounting commands. Five such commands exist:

`prtacct` Takes as input total accounting records and displays the records in ASCII readable format.

`acctmerg` Combines the contents of separate total accounting files into a single total accounting file. This command allows the merging of disk, process, and connect session total accounting records.

`runacct` Is the main accounting shell script. Normally invoked daily by cron, this command processes disk, connect session, process, and fee accounting information and produces summary files and reports. It accomplishes its task by proceeding through various states. In each successive state it invokes accounting commands to perform a specific task. For example, in one state, total accounting records for connect sessions are created; in another, disk, connect session, process, and fee total accounting records are merged to create one total accounting file.

`prdaily` Invoked by runacct to format a report of the previous day's accounting data; the report is in the file `/user/adm/acct/sum/rpt` *mmdd* where *mmdd* is the month and day of the report. runacct may also be used to display a report of the current day's accounting information.

monacct — Invoked once a month (or accounting period), this command summarizes daily accounting files and produces a summary files for the accounting period.

Login and directory structure

You now know the basics, but you still can't begin learning the day-to-day usage of accounting commands until you know where to log in. In addition, you should know the accounting directory structure; where the various commands, directories, and files are located. These two topics are discussed here.

Logging in

The login name for System Accounting is `adm`; the user ID for `adm` is 4. The adm login is a member of the user group adm; the group `adm also` has a group ID of 4.

The home directory for the adm login is `/usr/adm`. You log in to System Accounting the same way you do for any account; supply the login name to the SPP-UX login prompt:

```
login: adm
```

Note

The integrity of accounting data files must be maintained if System Accounting is to generate accurate reports. For this reason, it is highly recommended that a password be used with the `adm` login.

Directory structure

System Accounting uses a multi-level directory structure to organize its many accounting files. Each directory in this structure stores related groups of files, commands, or other directories. (See the section "System Accounting files" in this chapter for definitions of the accounting data files.)

The following directories are used by System Accounting:

- `/usr/adm` contains all active data-collection files, such as `pacct` and `fee`.
- `/usr/adm/acct` contains the `nite`, `sum`, and `fiscal` directories described below.

- /usr/adm/acct/nite stores data files that are processed daily by runacct.
- /usr/adm/acct/sum cumulative summary files updated by runacct are kept here.
- /usr/adm/acct/fiscal periodic (monthly) summary files created by monacct are stored here.
- /usr/lib/acct System Accounting commands reside here.
- /etc contains wtmp, and shell scripts rc and shutdown.

Disk space usage accounting

System Accounting provides the means to monitor disk space utilization for individual users. In this section, disk space usage accounting commands are explained.

Disk usage commands provide two main functions: they report disk usage (in blocks) for individual users and create disk total accounting records (supplied as inputs to commands such as prtacct or runacct).

Reporting disk space usage

Two commands, acctdusg and diskusg, report disk usage for individual users; both commands show the number of disk blocks allocated to specific users. However, each command has slightly different options. In addition, each differs in the manner in which it produces accounting information.

acctdusg

acctdusg takes from standard input a list of path names, usually created by the find command. For each file in the list, acctdusg identifies the owner of the file, computes the number of blocks allocated to the file, and adds this amount to a running total for the file's owner. When finished looking through the list, acctdusg displays the information accumulated for each user: user ID, user name, and number of blocks used.

This command is useful for reporting disk usage information for specific users or files. For example, suppose you want to know how many blocks of disk space you are using: your user ID is 351, user name is `bill`, and your home directory is `/users/pseudo/bill`. The following illustrates how you would use the `find` and `acctdusg` commands to show this information.

```
find /users/pseudo/bill -hidden -print > bills.files
acctdusg < bills.files
00351   bill                30
00351 bill 30 rm bills.files
```

In the above example, `bill` is using 30 blocks of disk space. The series of commands shown could easily have been combined into one line, such as:

```
find $HOME -hidden -print | acctdusg
00351   bill                30
```

The next example shows how to use `acctdusg` to generate disk usage information for all files in the system:

```
find / -hidden -print | acctdusg
00350 fred 11
00351 bill 30
00352 mike 17
00353 sarah 13
00354 molly 18
00000 root 3
00004 adm 36
00001 bin 2434
```

Two options are included with acctdusg:

`-u` *no_owners* — If -u is given, then path names of the files for which no owner is found are written into the file *no_owners*. This option could potentially find users who are trying to avoid disk charges.

`-p` *p_file* — The password file `/etc/passwd` is the default file used by acctdusg to determine ownership of files. If the `-p` option is used, then `acctdusg` will use *p_file*

instead. This option is not needed if your password file is /etc/passwd.

The shell script grpdusg, provided in the section "Sample Accounting Shell Scripts" later in this chapter, displays disk accounting information for users in a given group. It illustrates the use of the -u option with acctdusg.

diskusg

This command reports disk usage information in the same format as acctdusg; user ID, user name, and total disk blocks used. However, diskusg generates disk accounting information by looking through the inodes of a specified special file. (See the inode(4) man page for more information on inodes and special files.) Therefore, diskusg is faster and more accurate than acctdusg.

The syntax of the diskusg command is:

diskusg [*options*] [*files*]

It generates a disk usage report from data in *files*, if specified; otherwise standard input is used. diskusg is normally invoked with the *files* argument. When specified, *files* are the special file names of the devices containing the inode information used by diskusg to generate its report. *files* is normally a special file from the /dev directory.

The following options may be used with diskusg:

-s
: This tells diskusg that: (1) input is in diskusg output format, and (2) that all lines for a single user should be combined into a single line. This option is used to merge data from separate files, each containing the output from using diskusg on different devices.

-v
: This option is useful for finding users who are trying to avoid disk space accounting charges. When this option is specified, diskusg writes records to stderr (standard error output) showing the special file name, inode number, and user ID of files that apparently have no owner.

-i *fnmlist*
: Causes diskusg to ignore the data on those file systems whose file system name is in *fnmlist*. *fnmlist*

is a list of file systems separated by commas or enclosed within quotes.

-p *p_file* This is the same as the -p option of acctdusg.

-u *u_file* This option produces exactly the same output as the -v option. The difference between the two options is that -v writes its output to stderr; this option writes its output to the file *u_file*.

The output of diskusg is normally used by acctdisk to create disk total accounting records. In addition, diskusg is normally called by dodisk.

The following example creates disk usage information for all users whose files reside on the disk whose device file is /dev/rdsk/1s0. The file system used in this example is the same as was used in the previous acctdusg example.

```
diskusg /dev/rdsk/1s0
0 root 10616
1 bin 778
4 adm 96
350 fred 14
351 bill 32
352 mike 20
353 sarah 16
354 molly 22
355 julie 2
501 guest 2
```

The differences between diskusg and acctdusg are best illustrated by comparing their outputs:

1. acctdusg places leading zeros on user IDs; diskusg doesn't.
2. acctdusg counts files only under each user's $HOME directory. Files that users own in directories other than their home directory (for example, files in the /tmp directory) are counted as files with no owner.
3. Two extra users, julie and guest, show up in the output of diskusg when compared with the output from acctdusg. This occurred

because the directories of these two users were empty; therefore, no disk usage totals were generated by `acctdusg`. However, `diskusg` looked at inodes and saw that `julie` and `guest` were actually using two blocks for the directories themselves.

4. If two or more users have links to a particular file, then `acctdusg` will prorate disk space usage for the file between each user. For example, if three users had a link to a 300-block file called `skurbnich.dat`, each user would be charged for 100 blocks of this file.

Creating total accounting records

Two commands are used to create total accounting records: `acctdisk`, and `dodisk`.

acctdisk

`acctdisk` uses standard input records of the format produced by `acctdusg` and `diskusg`. From these records, `acctdisk` produces disk total accounting records that may be inputs to `prtacct` or `runacct`.

The following would write disk total accounting records to the file disktacct for all users in the group `pseudo`:

```
find / -group pseudo -print | acctdusg | acctdisk > disktacct
```

The next example would generate disk total accounting records for all users who have files on the disk `/dev/rdsk/1s0`. The total accounting records are written to the file `disktacct`.

```
diskusg /dev/rdsk/1s0 | acctdisk > disktacct
```

`acctdisk` has no options and is normally invoked by `dodisk`.

dodisk

`dodisk` is normally invoked by `cron` to create disk total accounting records for daily usage by System Accounting. The syntax for `dodisk` is:

```
dodisk [-o] [files...]
```

In the default case, `dodisk` creates disk total accounting records on the special files whose names are stored in `/etc/checklist`; the special file names are supplied as input to `diskusg`, which pipes its output to `acctdisk`, which in turn creates total accounting records.

If the `-o` option is used, `dodisk` creates total accounting records more slowly by using `acctdusg` instead of `diskusg`.

If *files* are used, disk accounting will be done on these file systems only. When the `-o` option is used, *files* should be mount points of mounted file systems; if omitted, *files* should be the special file names of mountable file systems.

See the "Installation and daily usage" section of this chapter for more information on how `dodisk` should be invoked by `cron`.

It is possible for malicious users to defeat disk space accounting by giving their files away to other users with chown(2) or chown(1) (by default, all users can execute them). To avoid this, take away the ability to use these commands from some or all users with the setprivgrp(1M) command. To let only the superuser use the change-ownership abilities, add the following line to `/etc/rc`:

```
setprivgrp -n CHOWN
```

To let one or more groups of users use the change-ownership abilities, add a line for each group to `/etc/rc`, similar to the following:

```
setprivgrp CHOWN
```

Taking away the change-ownership ability may cause problems when running some commands or applications.

Connect session sccounting

Whenever a user logs in to or out of SPP-UX, the program login records the connect session in the file `/etc/wtmp`. Records in wtmp contain the following information:

- the terminal name on which the connect session occurred
- the login name of the user
- the current time/date at login or logout
- other status information(see the utmp(4) man page for details)

System Accounting provides commands that allow you to write records to `wtmp`, to display and manipulate `wtmp`, and to create total accounting records from `wtmp`. These commands are covered in this section.

Writing records to wtmp (acctwtmp)

The command `acctwtmp` allows you to write records to `wtmp` for whatever reason you might have. `acctwtmp` is normally invoked by `startup` and `shutacct` to record when System Accounting was turned on and off, respectively. The format of the command is:

```
acctwtmp "reason"
```

where *reason* is a string describing the reason for writing the record to `wtmp`. `acctwtmp` does not directly write records to `wtmp`: it writes a record containing the terminal name, current time, and reason string to standard output. To actually write the record to `wtmp` you must append the output from `acctwtmp` to the `wtmp` file as follows:

```
acctwtmp "reason" >>/etc/wtmp
```

The *reason* string may be any combination of letters, numbers, spaces, and the dollar sign ($), but may not exceed 11 characters in length. (*reason* must be enclosed in double quotes as shown.)

Displaying connect ression records (fwtmp)

To display the contents of `wtmp`, you can use the command `fwtmp`. When no options are used, `fwtmp` uses standard input records of the format contained in wtmp; it writes to standard output the ASCII readable equivalent of the input records. Two alternatives exist for the output from this command:

- The output of this command can be edited, via an SPP-UX editor such as `vi`, and then rewritten to `wtmp` using special `fwtmp` options described below.
- The output can be supplied as input to commands which convert the information to total accounting records.

The syntax of `fwtmp` is:

```
fwtmp [-ic]
```

If no option is specified for the `fwtmp` command, then input is in binary format and is to be converted to ASCII readable format. The various combinations of the options `i` and `c` provide other combinations of input and output formats. The possible options are described below:

Option	Description
`-ic`	Input is in ASCII readable form and is to be converted to binary form. This is essentially the opposite of using `fwtmp` without any options.
`-i`	Both input and output are in ASCII readable format. This is the same as performing an ASCII to ASCII copy.
`-c`	Both input and output are in binary format; a binary-to-binary copy.

The following example shows the output produced by `fwtmp`. It is followed by a description of each column in the report:

```
fwtmp < /etc/wtmp
         system boot    0    2 0000 0000 479472540 Mar 12 03:49:00 1994
root  co  console       0    7 0000 0000 479475173 Mar 12 04:32:53 1994
         acctg on       0    9 0000 0000 479493135 Mar 12 09:32:15 1994
mike  a1  ttya1       352    7 0000 0000 479493590 Mar 12 09:40:00 1994
mike  a1  ttya1       352    8 0011 0000 479496000 Mar 12 10:20:00 1994
sarah 07  tty07       353    7 0000 0000 479518335 Mar 12 16:32:15 1994
bill  10  tty10       351    7 0000 0000 479521475 Mar 12 17:24:35 1994
sarah 07  tty07       353    8 0011 0000 479522478 Mar 12 17:41:18 1994
bill  10  tty10       351    8 0011 0000 479526487 Mar 12 18:48:07 1994
      co  console       0    8 0011 0000 479526488 Mar 12 18:48:08 1994
         acctg off      0    9 0000 0000 479526493 Mar 12 18:48:13 1994
         system boot    0    2 0000 0000 479389800 Mar 12 05:00:00 1994
```

Column	Description
1	The login name of the user who logged in or out.
2	`/etc/inittab` ID (this is usually the number of the line on which the connect session took place).
3	The name of the device on which the connect session occurred.
4	Process ID of the user who logged in or out.
5	Entry type. This field contains information on the type of record; for example, it shows whether the record is a login record (entry type=7), logout record (entry type=8), or if the record was written by `acctwtmp` (entry type=9). See the utmp(4) man page for more details on this field.
6-7	Exit status for connect session. See the login(1) l and utmp(4) man pages for details.
8	Time that entry was made (in elapsed seconds since January 1, 1970).
9-12	The equivalent of column 8 in date/time format showing month, day, time of day (in 24-hour format), and year.

Fixing wtmp errors (wtmpfix)

When a user logs into SPP-UX, the `login` program stores the value seven (7) in the entry type field of the connect session record. When the same user logs out, an entry type of eight (8) is recorded. You can see this by examining the sample output created by

fwtmp in the previous section. Note that in the example, login records precede their corresponding logout records in chronological order.

Occasionally, this time-stamped ordering becomes inconsistent: logout records might precede login records. (This occurs when the date and time are reset while users are still logged in.) When this happens, the commands that create connect session total accounting records will not work properly.

Fortunately, the command wtmpfix fixes corrupted wtmp files. wtmpfix takes wtmp binary records as input and corrects the time/date stamps to be consistent; its standard output is also binary wtmp records. The syntax for wtmpfix us:

```
wtmpfix [files]
```

If *files* is given, then input is taken from *files*. A dash (-) can be used in place of *files* to indicate standard input. If you specify wtmp as both input to and output from this command, wtmp will be destroyed. Therefore, take care not to destroy wtmp. The following shows how to properly fix wtmp using wtmpfix:

```
wtmpfix /etc/wtmp > wtmp.temp
fwtmp -c < wtmp.temp > /etc/wtmp
rm wtmp.temp
```

Creating Total Accounting Records

This final set of connect session accounting commands is used to create connect session total accounting records. Before reading any further, you may want to review (in the "System Data Flow" section of this chapter).

acctcon1

acctcon1 converts a sequence of login/logoff records (of the format contained in wtmp) read from its standard input to a sequence of records, one per login session. Its input is normally redirected from wtmp; its output is columnar ASCII and can be supplied as input to prctmp or acctcon2.

The use of acctcon1 is illustrated below by first displaying the contents of wtmp with fwtmp, and

then using acctcon1 to create a connect session summary file. acctcon1's columnar data acctcon1 is described following the report:

```
fwtmp < /etc/wtmp
          system boot    0    2 0000 0000 479472540 Mar 12 03:49:00 1994
root   co  console       0    7 0000 0000 479475173 Mar 12 04:32:53 1994
          acctg on       0    9 0000 0000 479493135 Mar 12 09:32:15 1994
mike   a1  ttya1        352    7 0000 0000 479493590 Mar 12 09:40:00 1994
mike   a1  ttya1        352    8 0011 0000 479496000 Mar 12 10:20:00 1994
sarah  07  tty07        353    7 0000 0000 479518335 Mar 12 16:32:15 1994
bill   10  tty10        351    7 0000 0000 479521475 Mar 12 17:24:35 1994
sarah  07  tty07        353    8 0011 0000 479522478 Mar 12 17:41:18 1994
bill   10  tty10        351    8 0011 0000 479526487 Mar 12 18:48:07 1994
       co  console       0    8 0011 0000 479526488 Mar 12 18:48:08 1994
          acctg off      0     0000 0000 479526493 Mar 12 18:48:13 1994
acctcon1 < /etc/wtmp
20095488 353   sarah 1665   2478    479518335 Tue Mar 12 16:32:15 1994
521012224 352  mike                 479493590 Tue Mar 12 09:40:00 1994
520095488 351  bill  0      5012    479521475 Tue Mar 12 17:24:35 1994
521011712 0    root  41047  6488    479475173 Tue Mar 12 04:32:53 1994
```

Column	Description
1	Shows the device address (in decimal equivalent of major/minor device address) at which the connect session occurred.
2	Gives the user ID for the connect session record.
3	Displays the login name for the user.
4	Shows the number of prime connect time seconds that were used during the connect session.
5	Shows non-prime connect seconds.
6	Displays the connect session starting time (in seconds elapsed since January 1, 1970).
7-11	Shows the conversion of column six to date/time format showing month, day time of day (in 24-hour format), and year.
	In addition to its normal usage, `acctcon1` has four options:

Option	Description
`-p`	This option tells `acctcon1` not to produce one record per connect session. Instead, `acctcon1` echoes its input, one line per `wtmp` record, showing line name, login name, and time (in both seconds and day/time format). Using this option is similar to using `fwtmp`, except that this option doesn't show status information, whereas `fwtmp` does.
`-t`	`acctcon1` maintains a list of lines on which users are logged in. When it reaches the end of its input, it emits a session record for each line that still appears to be active. It normally assumes that its input is a current file, so that it uses the current time as the ending time for each session in progress. The `-t` flag causes it to use, instead, the last time found in its input, thus assuring reasonable and repeatable numbers for non-current files.
`-l` *file*	This option causes a line usage summary report to be placed in *file*. This report shows each line's name, number of minutes used, percentage of total elapsed time used, number of sessions charged, number of logins, and number of logins and logoffs. This report can be used to keep track of line usage, identify bad lines, and find software/hardware oddities. Hang-up, termination of `login`, and termination of the login shell each generate logoff records; therefore, the number of logoffs is often three to four times the number of connect sessions.
`-o` *file*	Using the `-o` option (for example, `acctcon1 -o f_overall`) causes *file* to be filled with an overall record for the accounting period, giving starting time,

ending time, number of reboots, and number of date changes.

The following example of the line use file (`line_use`) is created from the same `wtmp` file used in the previous `acctcon1` example; the standard output of `acctcon1` has been redirected into the file `ctmp`:

```
acctcon -t -l line_use < /etc/wtmp > ctmp
cat line_use
TOTAL DURATION IS 899 MINUTES
LINE      MINUTES   PERCENT # SESS # ON    # OFF
console   856       95      1       1       1
tty07     69        8       1       1       1
ttya1     40        4       1       1       1
tty10     84        9       1       1       1
TOTALS    1049      --      4       4       4
```

prctmp

The `prctmp` command is simple. Its only function is to put headings on the output created by `acctcon1`. `prctmp` makes a readable report from the output of `acctcon1`.

`prctmp` takes its input from standard input; therefore, to create a `prctmp` report from `acctcon1` information, you can simply pipe the output from `acctcon1` into `prctmp` as follows:

```
acctcon1 < /etc/wtmp | prctmp
```

`prctmp` will respond by generating a report with appropriate headings over the columns of output from `acctcon1`.

acctcon2

`acctcon2` creates connect session total accounting records from standard input of the format created by `acctcon1`. In other words, to create connect session total accounting records, send the output from `acctcon1` into the input of `acctcon2`.

The total accounting records created by `acctcon2` are sent to standard output. So if you want to store these records, you must redirect standard output. The following command line shows how to write

total accounting records from the connect session record file (wtmp) into the file ctacct:

```
acctcon1 < /etc/wtmp | accton2 > ctacct
```

Process accounting

Process accounting commands provide the means to accumulate execution statistics, such as memory usage, CPU time, number of input/output transfers, for individual processes. This section describes how to:

- Turn process accounting on
- Turn process accounting off
- Make sure that the process accounting file (pacct) doesn't become too large
- Display process accounting records
- Generate a command summary report
- Create total accounting records from the process accounting file.

Turning process accounting on

Before System Accounting can generate process accounting data, process accounting must be turned on. Two commands can be used to accomplish this task: turnacct on and accton. After process accounting has been turned on, the kernel will write a process accounting record for every terminating process. The record will be written into the current process accounting file (pacct by default).

The startup command, placed in the system initialization shell script /etc/rc, automatically turns process accounting on. Therefore, if you have updated /etc/rc for System Accounting (as described in the section "How to Install System Accounting" in this chapter), process accounting will automatically be activated, and you should seldom need to use the commands described here.

These commands are described for your benefit in case you ever need to manually turn process accounting on or off.

turnacct on

The command used most often to activate accounting is `turnacct on`; only the superuser and the `adm` login can execute this command. `turnacct on` assumes that the process accounting file is the default file pacct. The action of `turnacct on` can be summarized as follows:

1. Check to see if the process accounting file `pacct` exists.
2. If pacct doesn't exist, then create a new `pacct` file.
3. Turn process accounting on by invoking accton with the filename argument `pacct`.

To execute this command, simply enter `turnacct on` at the SPP-UX prompt.

accton

Again, only the superuser and the `adm` login can execute `accton`. When invoked with a filename argument, `accton` turns on process accounting and makes the specified filename the current process accounting file. For example,

```
accton /usr/adm/pacct
```

tells the kernel to start writing process accounting records to the file called `/usr/adm/pacct`. The next example would activate process accounting and make the current process accounting file `/usr/adm/XX107`:

```
accton /usr/adm/XX107
```

The filename you specify must be an existing file; otherwise, `accton` will fail.

`accton` calls another routine, `acct`. `acct` is the system call that actually tells the kernel to start writing process accounting records. See the `acct(2)` man page for more information.

Turning process accounting off

Two commands are used to turn process accounting off: `turnacct off` and `accton` (with no filename argument). These commands tell the kernel to stop

writing records to the current process accounting file.

If you have updated the /etc/shutdown shell script as described in the section "How to Install System Accounting" in this chapter, you will seldom, if ever, use these commands. The reason is that the shutacct command, added to /etc/shutdown, automatically turns process accounting off.

turnacct off

turnacct off can be executed only by the superuser and the adm login. turnacct off turns process accounting off by invoking the accton command without the optional filename argument. You execute this command by typing:

```
turnacct off
```

accton

When accton is invoked without the optional filename argument, process accounting is turned off. You would enter this command as:

```
accton
```

accton tells the kernel to stop writing process accounting records by using the system call acct.

Checking the size of pacct

On a multiuser system, many processes can execute during a single hour. Therefore, process accounting files have the potential to become quite large. System Accounting has built-in mechanisms that insure that the default process accounting file pacct doesn't become too large. The two commands used for this purpose are: turnacct switch and ckpacct.

The commands described here work only on the default process accounting file, pacct.

ckpacct

The command ckpacct is normally invoked by cron every hour to insure that the current process accounting file pacct hasn't become to large. The format of ckpacct is:

ckpacct [*blocks*]

If the size of pacct exceeds the *blocks* argument, 1 000 by default if *blocks* is not specified, then turnacct switch is executed. turnacct switch renames the current pacct file and creates a new pacct file.

If the amount of free space falls below a certain threshold, ckpacct will automatically turn off process accounting via turnacct off.

The kernel may enforce a size limit on the size of pacct. This will take precedence over the limit set by ckpacct. See the acctsh(1M) and acct(2) man pages for more details.

turnacct switch

turnacct switch is used to create a new pacct file when the current pacct file is too large. The action of turnacct switch can be summarized as follows:

1. Process accounting is temporarily turned off.
2. The current pacct file is renamed to pacct *incr*, where *incr* is an integer starting at 1 and incrementing by one for each additional pacct file that is created via turnacct switch.
3. After the old pacct file is renamed to pacct *incr*, a new, current pacct file is created.
4. Process accounting is restarted; the kernel starts writing records to the newly created pacct file.

The example below illustrates the effect of using the turnacct switch command. In the example, turnacct switch is executed from the adm home directory /usr/adm. Comment lines begin with a cross-hatch (#) and are included in the example only as explanatory material:

```
#
# First, list all the process accounting files
# (at this point, there is only one).
#
ll pacct*
-rw-rw-r-- 1 adm adm 2196 Mar 21 12:44 pacct
#
# Now execute turnacct switch, which will rename the current
# pacct file to pacct1 and will create a new pacct file.
#
turnacct switch
#
# Now verify this by listing all process accounting
# files again.
#
ll pact*
-rw-rw-r-- 1 adm adm 72 Mar 21 12:46 pacct
-rw-rw-r-- 1 adm adm 2196 Mar 21 12:44 pacct1
#
# The current process accounting file is pacct. The previous
# process accounting file is now named pacct1.
#
```


Displaying process accounting records using acctcom

The acctcom command allows you to display records from any file containing process accounting records. Normally you would use this command to display records from the pacct files (`pacct`, `pacct1`, `pacct2` ...).

`acctcom` is a very versatile command; its syntax follows:

```
acctcom [ [options] [file] ]...
```

If no *file* is specified, `acctcom` uses the current `pacct` file as input. Input can also be taken from standard input. Some of `acctcom`'s options allow you to select only the records that you want to see; other options control the format of the report.

The information contained in this section is organized as follows:

- First, definitions are given for the columnar data produced by `acctcom`
- Command options that control the format of the report are discussed
- Options that allow you to select particular records are described
- Finally, to help you understand how to use `acctcom`'s options, sample `acctcom` reports are shown.

Definitions of information produced by acctcom

`acctcom` generates a columnar report with descriptive headings on each column. Each line of the report represents the execution statistics that a particular process accumulated during its lifetime. The standard columns in the report, that is, the columns that are displayed when none of `acctcom`'s options are specified, are shown below:

Column Header	Definition
`COMMAND NAME`	The name of the command or program that spawned the process is shown here. Whenever you enter a command, you are spawning a process. For example, if you enter the command `ll /usr/lib/acct` you are creating a process with the command name `ll`. If a command requiring superuser privileges is executed, a # appears before the command name.
`USER`	The login name of the user who created the process is displayed here.
`TTYNAME`	This is the name of the terminal from which the process was executed. If the process was not executed from a known terminal (for example, if it

	was executed via cron), then a question mark(?) appears in this column.
START TIME	The time that the process began executing (in *hh:mm:ss* format) is displayed here.
END TIME	This is the time (*hh:mm:ss*) that the process finished executing.
REAL (SECS)	The number of seconds that elapsed from START TIME to END TIME is shown in this column.
CPU (SECS)	This column shows how much of the CPU's time a process used during its execution.
MEAN SIZE(K)	This is a rough estimate (in kilobytes) of the amount of memory that a process used during execution. This estimate is determined from the current process's memory usage at each system clock interrupt. It is, therefore, subject to statistical sampling errors. Only the memory resident pages of a process are counted; no pages in the swap space are counted. Shared code and data is divided among the processes using it. The size is divided by the number of processes sharing the code or data.

Listed below are the columns that are not displayed on the standard report, but which can be displayed by using acctcom options:

Column Header	Definition
F	For a process created by fork which does not do an exec, this column takes the value 1; commands that require superuser privileges show a 2; superuser commands that do a fork without an exec show a 3; otherwise, this column shows a 0.
STAT	This column displays the system exit status. (This is not the status returned by exit to a parent process during wait). When a process terminates normally, this field shows a <emph I 0 I. If a command terminates abnormally, then a value other than zero is shown. For example, if you interrupt a command with the DEL key, this column will contain a 2.
HOG FACTOR	The hog factor is computed as the CPU time divided by REAL time; it provides a relative measure of the available CPU time used by the process during its execution. For example, a hog factor of less than 0.50 indicates that the process spent less than half of its

time using the CPU. A hog factor of 0.75 indicates that a process spent 75% of its time using the CPU.

`KCORE MIN` — This calculation provides a combined measurement of the amount of memory used (in kilobytes) and the length of time it was used (in minutes). It is computed as follows:

```
KCORE MIN = CPU (SECS)MEAN SIZE(K) / 60
```

`CPU SYS` — This is the portion of total CPU time that was spent executing operating system code, such as system calls (for example, writing to disk).

`USER (SECS)` — This is the remaining portion of CPU time. User CPU time is the amount of time actually spent executing a process's code (rather than system code).

`CPU FACTOR` — Whenever you execute a command, the CPU spends part of its time actually executing the command's code (user CPU time) and spends the rest of its time performing system functions, such as writing to the disk or terminal (system CPU time). That is, total CPU time is comprised of both CPU SYS and USER CPU time:

```
CPU (SECS) = CPU SYS + USER (SECS)
```

The CPU factor shows the ratio of user CPU time to total CPU time:

```
CPU FACTOR = USER (SECS) / (CPU SYS + USER (SECS))
```

For example, if a command has a CPU factor of 0.35, that means that the CPU spent 35% of its time executing user code and 65% performing system functions.

`CHARS TRNSFD` — The number of characters (bytes) read and/or written by the command is displayed in this column.

`BLOCKS R/W` — This column shows the number of file system blocks read and/or written as a result of executing this command. This number is not directly related to `CHARS TRNSFD` and may vary each time the command is executed, because `BLOCKS R/W` is affected by directory searches made before opening files, other processes accessing the same files, and general file system activity.

Report format options

When no report format options are specified, acctcom will produce a report containing only the default information. Optional information can be displayed only by using the report format options. Definitions of the report format options follow:

Option	Description
-a	Cause average statistics to be displayed at the end of the report. The following information is shown: • total number of commands processed (cmds=xxx) • average real time per process (Real=x.xx) • average CPU time per process (CPU=x.xx) • average USER CPU time per process (USER=x.xx) • average SYS CPU time per process (SYS=x.xx) • average characters transferred (CHARS=x.xx) • average blocks transferred (BLK=x.xx) • average CPU factor (USR/TOT=x.xx) • average HOG factor (HOG=x.xx)
-b	Display the process records in reverse order: most recently executed commands will be shown first.
-f	Print the fork/exec flag (F column) and process exit status (STAT column) on the report.
-h	Cause the optional HOG FACTOR column to be displayed, instead of the standard mean memory size column MEAN SIZE(K).
-i	Replace the standard MEAN SIZE(K) column in the report with the optional I/O countsm CHARS TRNSFD and BLOCKS R/W.
-k	Replace the standard MEAN SIZE(K) column with KCORE MIN.
-m	Show the default column MEAN SIZE(K) on the report. This option is used to include MEAN SIZE(K) when it has been bumped off by another option. The following example produces a report showing both KCORE MIN and MEAN SIZE(K):

```
acctcon -km
```

`-r`	Include the optional `CPU FACTOR` column in the report.
`-t`	Show separate system and user CPU times (`CPU SYS` and `USER (SECS)`, respectively).
`-v`	Suppress the printing of column headings at the top of the report.
`-q`	This option is the same as the `-a` option, except that individual process accounting records are not displayed; only the averages are displayed.
`-o` *ofile*	Copy the input process accounting records to *ofile*.

Record Selection Options

The options described here allow you to select the records that are included in the report produced by `acctcom`. For each option, descriptions and examples are provided:

Option	Description
`-l` *line*	Display only the processes that were executed from the user terminal `/dev/`*line*. For example:

```
acctcom console
```

would display records only for the processes that were created from the terminal console.

`-u` *user*	Show only the processes belonging to *user*. *user* can be any of the following:

- a user ID (for example, `acctcom -u 355`)
- user name (`acctcom -u julie`)
- a cross-hatch (#) (`acctcom -u#`)
- a question mark (?) (`acctcom -u?`)

If # is specified as the user name, then only the commands that require superuser privileges will be displayed by `acctcom`. If ? is given as the user, then only the processes with unknown process IDs will be displayed. As an example, the following two commands are equivalent:

```
acctcon -u 0

acctcom -u root
```

`-g` *group* Show only the processes belonging to *group*. *group* may be specified as either a group name or group ID. For example, suppose the group `pseudo` with group ID 300 is defined in `/etc/group`; then the following two commands are equivalent:

```
acctcon -g 300

acctcom -u root
```

`-s` *time* Select processes existing at or after *time*. Time is given in 24-hour format: *hr*[:*min*[:*sec*]]. The following example would display all the processes that existed at or after 3:30pm:

```
acctcom -s 15:30
```

`-e` *time* Select processes that existeat or before *time*. Time is supplied in 24-hour format: *hr*[:*min*[:*sec*]]. The next example would display all the processes that existed between midnight and 12:15am:

```
acctcom -e 0:15
```

`-S` *time* Select processes starting at or after time where *time* is in 24-hour format. The following example would display all the processes that started at 1:30:42pm or after:

```
acctcom -S 13:10:42
```

`-E` *time* Display only the processes that terminated at or before *time*, where time is in 24-hour format. Both the `-S` and `-E` options with the same *time* argument will cause `acctcom` to display only the processes that existed at the specified time. For example, to see all the processes that existed at exactly 30 minutes past noon, you would enter:

```
acctcom -S 12:30 -E 12:30
```

`-n` *pattern* Show only the commands matching *pattern*. *pattern* can be a regular expression as described in the `ed(1)` man page, except that + means one or more occurrences. For example, to display all processes that were created by executing the `ls` command, you would enter:

```
acctcom -n ls
```

To display all the commands that start with acct, enter:

```
acctcom -n acct
```

To see all the commands that contain the letter m in their spelling, you can use the wild card character *. Type:

```
acctcom -n .*m.*
```

`-H` *factor* Display only those processes whose hog factor exceeds *factor*. For example,

```
acctcom -H 0.85
```

would display all the processes that spent over 85% of their execution time in the CPU. You can use this option to find greedy processes, processes that are hogging the CPU.

`-O` *time* Show only those processes whose system CPU time exceeds *time*, specified in seconds. The following example would be used to determine which processes took more than 8.25 seconds of operating system CPU time to execute:

```
acctcom -O 8.25
```

This option could be used to determine which processes are making heavy use of the operating system calls.

`-C` *sec* Show only the processes whose total CPU time (`SYS` + `USER`) exceeds *sec* seconds. The next example would display all the processes that used over 5.28 seconds of CPU time to execute:

```
acctcom -C 5.28
```

`-I` *chars* Display only the processes transferring more characters than the limit given by *chars*. For example,

```
acctcom -I 10240
```

will display all the processes that transferred over ten kilobytes of characters (10 240 = 10 * 1 024 bytes).

Command summary report (acctcms)

The `acctcms` command takes process accounting records as input; but instead of reporting on the individual processes, `acctcms` generates a report

on the commands that generated the process accounting records. The action of `acctcms` can be summarized as follows:

1. `acctcms` looks through the input process accounting records and accumulates execution statistics for each unique command name. This information is stored in internal summary format, one record per command name.
2. Depending on the `acctcms` options used, the command summary records created in step 1 are sorted.
3. The command summary records are written to standard output in the internal summary format mentioned in step 1. This format is not readable.

The syntax of the acctcms command is:

```
acctcms [options] files
```

where *files* is a list of the input process accounting files for which the command summary report is to be generated. The *options* are discussed in the following sections.

Producing a readable report (-a option)

By default, the output of `acctcms` is in internal summary record format; if you display it at your terminal, it will be unreadable. To get a readable ASCII report, use the `-a` option.

The `-a` option causes acctcms to produce a report with descriptive column headings. Total and average (mean) execution statistics for each command are displayed, one line per command, along with total and average statistics over all commands in the report. The columnar data produced by `acctcms` is as follows.

Column Header	Description
COMMAND NAME	The name of the command for which execution statistics are summarized. All shell procedures are grouped together under the name `sh`, because only object modules are reported by the process accounting system.
NUMBER CMDS	The total number of times that the command was invoked.

TOTAL KCOREMIN — The total amount of kcore minutes accumulated for the command. (See the section "Definitions of information produced by acctcom" in this chapter for a more complete description of kcore minutes.)

TOTAL CPU-MIN — The total CPU time that the named command has accumulated.

TOTAL REAL-MIN — Total accumulated real time minutes are displayed in this column.

MEAN SIZE-K — The average amount of memory (in kilobytes) consumed by the command.

MEAN CPU-MIN — The average CPU time consumed per command invocation is shown here; the following equation shows how it is computed:

```
MEAN CPU-MIN = TOTAL CPU-MIN / NUMBER CMDS
```

HOG FACTOR — The average hog factor over all invocations of the command. It is computed as:

```
HOG FACTOR = TOTAL CPU-MIN / TOTAL REAL-MIN
```

CHARS TRNSFD — The total number of characters transferred by the command. Note that this number may sometimes be negative.

BLOCKS READ — A total count of the physical blocks read and written by the given command. (See the section "Displaying process accounting records (acctcom)" in this chapter for details on the significance of this total.)

When only the `-a` option is specified, the report is sorted in descending order on the TOTAL KCOREMIN column: commands using more TOTAL KCOREMIN are shown before those using fewer TOTAL KCOREMIN. This report gives a relative measure of the amount of memory used over time by the various commands: commands toward the start of the report are making more use of memory resources than are commands toward the end of the report.

Other options

In addition to the `-a` option, several other options can be used to control the format of the report generated by `acctcms`. Some options specify which field to sort the report on; other options control the printing of prime/non-prime time usage. The options and a description of their use follow:

Option	Description
`-c`	Sort the commands in descending order on `TOTAL CPU-MIN`, rather than the default `TOTAL KCOREMIN`. This report can be used to determine which commands are using most of the computer's CPU time.
`-n`	Cause the report to be sorted in descending order on the column named `NUMBER CMDS`. Commands toward the start of this report are the ones used most frequently; commands toward the end are used least often.
`-j`	Combine all commands invoked only once on one line of the report. This line is denoted by having "***other" in the `COMMAND NAME` column. This option is useful for shortening a report that has many one-invocation commands.
`-o`	Used only with the `-a` option, `-o` causes the report to be generated only for commands that were executed during non-prime time (as specified in the holidays file). You can use this option to get a non-prime time command summary report.
`-p`	Also used only with the `-a` option, `-p` elicits a report generated only for commands that were executed during prime time (as specified in holidays). This option is used to get a prime time command summary report.
`-apo`	When the options -o and -p are used together with -a, a combination prime and non-prime time report is produced. The output of this report is same as that produced by -a alone, except that the `NUMBER CMDS`, `TOTAL CPU-MIN`, and `TOTAL REAL-MIN` columns are divided into two columns; one for prime time totals, the other for non-prime time. (Prime time columns have a (P) header, while non-prime time columns are headed by (NP).)
`-s` *files*	Specifies that any named input *files* following the `-s` on the command line are already in internal summary format. This option is useful for merging previous acctcms reports with current reports. The following example uses `-s` to create a command summary report from previous process accounting files (`pacct?`) and the current process accounting file (`pacct`). The final ASCII report is stored in the file `ascii_summary`.

```
acctcms pacct? > old_summary
acctcms pacct > new_summary
acctcms -as old_summary new_summary > ascii_summary
```

Sample report

The ASCII reports produced by `acctcms` contain more than 80 characters per line. When these reports are displayed at an 80-column terminal, the lines wrap around on the screen. In addition, if the report is printed on an 80-column printer, some of the rightmost columns will be lost. Therefore, be sure to do one of the following:

- Use a printer with compressed print capabilities, so that all of the report will fit on standard computer paper
- Use a printer with enough columns to display all the information, for example, a 132-column printer

The following example generates a command summary report for the current process accounting file (no file is specified, so the current `pacct` file is assumed). By giving the `-j` option, all the commands that were executed only once are grouped under the command name `***other`. Total execution statistics for all commands are grouped under the command name TOTALS.

```
acctcms -aj
```

TOTAL COMMAND SUMMARY

COMMAND NAME	NUMBER CMDS	TOTAL KCOREMIN	TOTAL CPU-MIN	TOTAL REAL-MIN	MEAN SIZE-K	MEAN CPU-MIN	HOG FACTOR	CHARS TRNSFD	BLOCKS READ
TOTALS	61	17.63	0.38	164.49	46.25	0.01	0.00	104553	1027
acctcms	17	12.13	0.16	0.35	76.72	0.01	0.45	49192	306
sh	8	2.43	0.09	152.86	26.79	0.01	0.00	9043	163
more	3	0.73	0.02	10.50	31.00	0.01	0.00	21618	83
ll	6	0.61	0.04	0.11	16.50	0.01	0.33	5715	95
acctcom	4	0.58	0.02	0.07	28.50	0.01	0.30	15319	42
***other	9	0.54	0.02	0.14	25.26	0.00	0.16	459	161
cat	4	0.19	0.01	0.35	22.97	0.00	0.02	3112	52
rm	2	0.11	0.00	0.02	22.22	0.00	0.29	0	29
chmod	2	0.10	0.00	0.01	22.00	0.00	0.35	0	15
accton	2	0.08	0.00	0.02	19.00	0.00	0.29	0	22
sed	2	0.08	0.01	0.04	14.50	0.00	0.13	73	38
echo	2	0.05	0.00	0.02	20.00	0.00	0.16	22	21

Creating total accounting records

Two commands, `acctprc1` and `acctprc2`, are used to create total accounting records from the process accounting files. The output from `acctprc1` is supplied as input to `acctprc2` which produces the total accounting records. These commands are normally invoked by runacct to produce daily accounting information.

acctprc1

This command reads process accounting records from standard input, adds login names corresponding to the user ID of each record, and then writes for each process an ASCII line showing:

- the ID of the user that created the process
- the user's login name
- prime CPU time in ticks (a "tick" is one fiftieth of a second)
- non-prime CPU time, also in ticks
- mean memory size (in pages, 4 Kbytes per page)

The format of `acctprc1` is:

```
acctprc1 [ctmp]
```

where *ctmp* contains a list of login sessions of the form created by `acctcon1`, sorted by user ID and login name.

The number of sessions should be 1000 or less. If there are more than 1000 sessions, the accounting system "hangs" (suspends indefinitely) and must be killed manually via the `kill` command and restarted.

To use `acctprc1`, input must be redirected from a process accounting file. The following example creates a file, `ascii_ptacct`, containing ASCII process accounting information that can be used to create process total accounting records. This file is created from the current process accounting file `pacct`.

```
acctprc1 <pacct >ascii_ptacct
```

Normally, `acctprc1` gets login names from the password file `/etc/passwd`, which is sufficient on systems where each user has a unique user ID. However, on systems where different users share the same user ID, the ctmp file should be specified; it helps `acctprc1` distinguish different login names that share the same user ID.

acctprc2

This command reads from standard input records of the form created by `acctprc1`; it then summarizes the records by user ID and name, and writes the sorted summaries to standard output as total accounting records. The following example creates total accounting records for all processes in the current process accounting file `pacct`; the total accounting records are stored in the file `ptacct`.

```
acctprc1 <pacct |acctprc2 >ptacct
```

Charging fees to users (chargefee)

System Accounting provides the capability to charge fees to specific users; the chargefee command is used to accomplish this task. `chargefee` allows you to charge generic *units* to a specific login name. The syntax of this command is:

`chargefee` *login_name number*

where *number* is the number of units to be charged to a particular user, and *login_name* is the login name of the user to whom *number* units are to be charged.

number can be any whole number in the range -32 768 to 32 767; when charging fees, keep in mind that the sum of each user's fees must also be within this range.

`chargefee` accumulates fee charge records in the file `/usr/adm/fee`. These records are then merged with other accounting records via runacct.

The following example charges 25 units to the user whose login name is `julie`:

```
chargefee julie 25
```

Suppose you inadvertently charged 247 units to the user named `zooey`, and you want to return her charges to their original value. You would enter the following:

```
chargefee zooey -247
```


Summarizing and reporting accounting information

This final group of commands summarizes and reports accounting information. Certain commands display and merge total accounting files, while others generate the daily and monthly reports used to analyze system performance and bill users for resource usage. The following commands are discussed in this section:

`prtacct`	displays total accounting records
`acctmerg`	merges total accounting files
`runacct`	generates daily summary files and reports
`prdaily`	displays the daily summary files and reports created by `runacct`
`monacct`	creates monthly summary files and reports

Displaying total accounting records (prtacct)

The prtacct command allows you to display the contents of a process accounting file. Its format is

```
prtacct file "heading"
```

where:

file is the name of the total accounting file to be displayed.

"heading" is a comment to be included in the standard report header produced by prtacct.

The format of the `prtacct` report is described next and is followed by an example.

prtacct report format

`prtacct` produces a columnar report with one line per total accounting record. Descriptive column headings are included in the report. Definitions of each column follow:

Column Header	Description
`UID`	This column shows the user ID of the owner of the total accounting record; that is, the ID of the user for whom the total accounting record was created.
`LOGIN NAME`	The login name of the owner of the total accounting record is displayed here.
`CPU (MINS)`	This column shows the total amount of CPU time (in minutes) that the user has consumed. This column is divided into prime and non-prime columns (`PRIME` and `NPRIME`, respectively). Information in these columns is created through process accounting commands.
`KCORE-MINS`	This represents a cumulative measure of memory and CPU time that a user consumed (See the section "Definitions of information produced by acctcom" in this chapter for a more precise definition). Information in this column is also divided into `PRIME` and `NPRIME` columns. This information is created through process accounting commands.
`CONNECT (MINS)`	This identifies the real time used (in minutes). In essence, what this column identifies is the amount of time that the user was logged in to the system. This

column is also subdivided into PRIME and NPRIME columns. The connect session accounting commands are the source of this information.

DISK BLOCKS — The total number of disk blocks allocated to the user is shown here. This information is created via disk space accounting commands.

OF PROCS — The total number of processes spawned by the user is displayed here. This information is created via the process accounting commands.

OF SESS — This column shows how many times the user logged in. Connect session accounting commands create this data.

DISK SAMPLES — This column indicates how many times the disk accounting was run to obtain the average number of disk blocks listed in the DISK BLOCKS column.

FEE — The number of fee units charged via `chargefee` is displayed here.

prtacct example

The following example displays disk total accounting records. First, the total accounting records are created via disk space accounting commands; then, they are displayed using `prtacct`. When examining this report, take note of the following:

- There are many similarities between this and the sample report produced by diskusg (see the section "Disk space usage accounting" in this chapter).
- Only the columns relating to disk space usage have non-zero values, because the total accounting records were created only from disk space usage accounting commands.

The example report produced by `prtacct` follows:

```
for file_system in 'cat /etc/checklist'
    do
    diskusg $file_system >dump.'basename $file_system'
    done
diskusg -s dtmp.* | sort +0n +1 | acctdisk >diskacct
prtacct diskacct "DISK TOTAL ACCOUNTING RECORDS"

Mar 26 17:01 1994 DISK TOTAL ACCOUNTING RECORDS Page 1
    LOGIN CPU   (MINS)  KCORE-MINS CONNECT(MINS)  DISK  # OF  # OF  # DISK FEE
UID NAME  PRIME NPRIME PRIME NPRIME PRIME NPRIME BLOCKS PROCS SESS SAMPLES
0   TOTAL 0     0     0     0     0     0     11598 0     0     10    0
0   root  0     0     0     0     0     0     10616 0     0     1     0
1   bin   0     0     0     0     0      0    778   0     0     1     0
4   adm   0     0     0     0     0     0     96    0     0     1     0
350 fred  0     0     0     0     0     0     14    0     0     1     0
351 bill  0     0     0     0     0     0     32    0     0     1     0
352 mike  0     0     0     0     0     0     20    0     0     1     0
353 sarah 0     0     0     0     0     0     16    0     0     1     0
354 molly 0     0     0     0     0     0     22    0     0     1     0
355 julie 0     0     0     0     0     0     2     0     0     1     0
501 guest 0     0     0     0     0     0     2     0     0     1     0
```

Merging total accounting files (acctmerg)

Normally executed by `runacct`, the `acctmerg` command merges separate total accounting files into a single total accounting file. All the total accounting records for a particular user name and ID are merged together to form one total accounting record for the given user name and ID. This command is useful for merging disk, connect session, and process total accounting files together to form a single, comprehensive total accounting file.

`acctmerg` reads standard input and up to nine additional files, all in total accounting record format. Its syntax is:

`acctmerg` [*options*] [*file*] . . .

where:

- *options* control the report format and the manner in which records are merged.
- *file* is one of up to nine files (in addition to standard input) that are to be merged into a single total accounting file, written to standard output.

Command options

The following options may be used with acctmerg to control the report format and the manner in which the total accounting records are merged:

Option	Description
`-a`	`acctmerg` normally produces output as total accounting records. The `-a` option causes `acctmerg` to produce output in ASCII. Note that the output generated by using this option is the same as the report produced by `prtacct`, except that no report headings or totals are displayed; only the columnar data is shown.
`-i`	In the default case, `acctmerg` assumes that its input files contain total accounting records. If `-i` is specified, then `acctmerg` will expect input files to be in the ASCII format created by the `-a` option.
`-p`	This option simply echoes input records; no merging or processing is done. The output is displayed in the format produced by the `-a` option.
`-t`	This option produces a single total accounting record that summarizes all input records. To see the ASCII version of this record, you must use the `-t` and `-a` options together:

```
acctmerg -t -a <tot_acct_recs
```

`-t` and `-a` can be specified in any order, but they must be specified separately as shown.

`-u`	Normally, `acctmerg` merges records that have the same user ID and user name. Using `-u` causes `acctmerg` to merge records on the basis of same user ID only; that is, the user name is disregarded as a key on which to merge records.
`-v`	This option causes `acctmerg` to produce output in verbose ASCII format. The same report is produced as the `-a` option, except that floating point numbers are displayed in more precise notation:

<mantissa>e<exponent>

Use the `-a`, `-v`, and `-i` options to edit total accounting records. For example, if you created a total accounting file (`ptacct`) containing process total accounting records, and you want to make some adjustments to these records, use the following sequence to "repair" this file:

```
acctmerg -v -a <ptacct >ptacct.ascii
```

next, edit `ptact.ascii` *as desired*

```
acctmerg -i <ptacct.ascii >ptacct
```

Example

The following example creates disk, process, and connect session total accounting records, merges them together, and stores the merged file in the file `merged_file`:

```
for fs in 'cat /etc/checklist
   do
   diskusg $fs >dtmp.'basename $fs'
   done
diskusg -s dtmp.* | sort +0n +1 | acctdisk >dtacct
acctcon1 <etc/wtmp | acctcon2 >ctacct
ptacct
for p_file in pacct*
   do
   acctprc1 <$p_file | acctprc2 >>ptacct
   donr
acctmerg dtacct ctacct <ptacct >tacct
```

Creating daily accounting information (runacct)

`runacct` is the main daily accounting shell procedure. Start `runacct` via `cron` during non-prime hours, when users are logged off. This is because it does not correctly log time for users that log on before running `runacct`.

runacct processes disk, connect session, process, and fee accounting files. It prepares cumulative summary files for use by prdaily and for billing purposes. This section discusses the following aspects of runacct:

- files processed by runacct
- the states that runacct progresses through while executing
- recovery from runacct failure
- restarting runacct
- reports produced by runacct

Files processed by runacct

The following files, processed by runacct, are of particular interest to the reader. (Filenames are given relative to the directory /usr/adm/acct.)

- nite/lineuse contains usage statistics for each terminal line on the system. This report is especially useful for detecting bad lines. If the ratio of logoffs to logins on a particular line exceeds 3 to 1, then there is a good possibility that the line is failing.
- nite/daytacct contains total accounting records from the previous day.
- sum/tacct contains accumulated total accounting records for each day's total accounting records (nite/daytacct) and can be used for billing purposes. It is restarted each month or fiscal period by the monacct shell script.
- sum/daycms is produced by acctcms. It contains the daily command summary. The ASCII version of this file is in nite/daycms.
- sum/cms holds the accumulation of each day's command summaries (sum/daycms). A new sum/cms file is created each month by monacct. The ASCII version of this file is in nite/cms.
- sum/loginlog maintains a record of the last time each user logged in.
- **sum/rprt** *mmdd* is the main daily accounting report created by runacct. The name for this

report is created automatically by the system with *mm* being the month and *dd* the day of the report. This report can be printed via `prdaily`.

`runacct` takes care not to damage files in the event of errors. A series of protection mechanisms are used that attempt to recognize errors, provide intelligent diagnostics, and terminate processing in such a way that `runacct` can be restarted with minimal intervention. To accomplish these goals, the following actions are performed by `runacct`:

- `runacct`'s progress is recorded by writing descriptive messages to the `nite/active` file.
- All diagnostics output during the execution of runacct are redirected to the file `nite/fd2log`.
- If the files `lock` and `lock1` exist when runacct is invoked, an error message will be displayed, and execution will terminate.
- The `lastdate` file contains the month and day that `runacct` was last run and is used to prevent more than one execution per day.
- If `runacct` detects an error, a message is written to `/dev/console`, mail is sent to `root` and `adm`, locks are removed, diagnostics files are saved, and execution is terminated.

The states of runacct

In order to allow runacct to be restartable, processing is broken down into separate re-entrant *states*. As runacct executes, it records its progress by writing the name of the most recently completed state into the file called `/usr/adm/statefile`. After processing for a state is complete, `runacct` examines statefile to determine which state to enter next. When runacct reaches the final state (`CLEANUP`), the `lock` and `lock1` files are removed, and execution terminates. The following are `runacct`'s states:

State	Action
`SETUP`	The command `turnacct` switch is executed. The process accounting files, `pacct?`, are moved to `Spacct?.`*mmdd*. The `/etc/wtmp` file is moved to `nite/wtmp.`*mmdd* with the current time added on the end.

WTMPFIX	nite/wtmp.*mmdd* is checked for correctness by wtmpfix. Some date changes will cause acctcon1 to fail, so wtmpfix attempts to adjust the time stamps in the nite/wtmp.*mmdd* file if a date change record appears.
CONNECT1	Connect session records are written to ctmp. The lineuse file is created, and the reboots file, showing all of the boot records found in nite/wtmp.*mmdd*, is created.
CONNECT2	ctmp is converted to connect session total accounting records in the file ctacct.*mmdd*.
PROCESS	The acctprc1 and acctprc2 programs are used to convert the process accounting files, Spacct?.*mmdd*, to the total accounting records in ptacct?.*mmdd*. The Spacct and ptacct files are correlated by number so that if runacct fails, the unnecessary reprocessing of Spacct files will not occur. One precaution should be noted: when restarting runacct in this state, remove the last ptacct file; if you don't, runacct will not finish.
MERGE	Merge the process and connect session total accounting records to form nite/daytacct.
FEES	Merge in any ASCII tacct records from the file fee into nite/daytacct.
DISK	On the day after the dodisk shell script runs, merge nite/disktacct with nite/daytacct.
MERGETACCT	Merge nite/daytacct with sum/tacct, the cumulative total accounting file. Each day, nite/daytacct is saved in sum/tacct*mmdd*, so that sum/tacct can be recreated in the event it becomes corrupted or lost.
CMS	Merge in today's command summary with the cumulative summary file sum/cms. Produce ASCII and internal format command summary files.
USEREXIT	Any installation-dependent (local) accounting programs can be run in this state. For example, you might want to execute commands that generate daily billing data for individual users (the shell script acct_bill in the section "Sample System Accounting Shell Scripts" could be used for this purpose). To have local accounting programs executed by runacct, enter the commands in

runacct in the code for the USEREXIT state of runacct.

CLEANUP — Clean up the temporary files, run prdaily and save its output in the file sum/rprtmmdd, remove the locks, then exit.

Recovering from failure

It is possible that runacct might fail and terminate abnormally. The primary reasons for runacct failure are:

- a system crash
- not enough disk space remaining in /usr
- a corrupted wtmp file

If the nite/active*mmdd* file exists, check it first for error messages. If the nite/active file and lock files exist, check fd2log for any mysterious messages. The following are error messages produced by runacct and the recommended recovery actions:

ERROR: locks found, run aborted

The files lock and lock1 were found. These files must be removed before runacct can be restarted.

ERROR: acctg already run for *date*: check /usr/adm/acct/nite/lastdate

The date in lastdate and today's date are the same. Remove lastdate before restarting runacct.

ERROR: turnacct switch returned rc=<emph|?|

Check the integrity of turnacct and accton. The accton program must be owned by root and have the setuid bit set.

ERROR: Spacct?.*mmdd* already exists

File setups have probably already run. Check the status of files, then run setups manually.

ERROR:/usr/adm/acct/nite/wtmp.*mmdd* already exists,run setup manually

You must perform the SETUP step manually, because the daily wtmp file already exists.

ERROR: wtmpfix errors see /usr/adm/acct/nite/wtmperror

wtmpfix detected a corrupted wtmp file. See the section "Fixing corrupted files" in this chapter for details on fixing wtmp errors.

ERROR: connect acctg failed: check /usr/adm/acct/nite/log

acctcon1 encountered a bad wtmp file. See the section "Fixing corrupted files" in this chapter for information on how to fix the file.

ERROR: Invalid state, check /usr/adm/acct/nite/active

The file statefile is probably corrupted. Check statefile and read active before restarting.

Restarting runacct

runacct is normally run via cron only once per day. However, if an error occurs while executing runacct (as described above), it may be necessary to restart runacct. runacct has the following syntax:

runacct [*state*]

runacct assumes that it is being invoked for the first time on the current day. The entry point for processing is based on the contents of statefile. To override statefile, include the desired entry *state* on the command line.

For example, to start runacct, you would enter:

```
nohup runacct 2> /usr/adm/acct/nite/f2dlog &
```

To restart runacct at state WTMPFIX:

```
nohup runacct WTMPFIX 2> /usr/adm/acct/nite/f2dlog &
```

All the above examples were run in the background (&) and use the nohup command so the process continues running even though you may log out.

Daily reports

runacct generates five basic reports upon each invocation. Brief descriptions of each report follow.

Daily Line Usage Report — Summarizes connect session accounting since the last invocation of runacct. It provides a log of system reboots, power failure recoveries, and any other records dumped into /etc/wtmp via acctwtmp. In addition, it provides a breakdown of line utilization.

Daily Resource Usage Report — Gives a summary of resource usage per individual user: it basically merges all the total accounting records for individual users and displays the records, one per user.

Daily Command Summary

Summarizes resource usage data for individual commands since the last invocation of `runacct`. The data included in this report is useful in determining the most heavily used commands; you can use these commands' characteristics of resource utilization when tuning your system.

This report is sorted by `TOTAL KCOREMIN`, an arbitrary but useful yardstick for calculating "drain" on a system.

Monthly Total Command Summary

This report is exactly the same as the Daily Command Summary, except that the Daily Command Summary contains command summary information accumulated only since the last invocation of `runacct`, while the Monthly Total Command Summary summarizes commands from the start of the fiscal period to the current date. In other words, the monthly report reflects the data accumulated since the last invocation of `monacct`.

Last Login

Gives the date each user last logged in to the system. This could be a good source for finding likely candidates for the archives, or getting rid of unused login directories.

Displaying runacct reports (prdaily)

As runacct finishes executing, it deposits a report of the current day's accounting in the file /usr/adm/acct/sum/rptmmdd, where *mmdd* is the month and day that the report was generated. The prdaily command is used to display the contents of any daily report file created by runacct. Its syntax is:

```
prdaily [-l] [-c] [mmdd]
```

where:

- *mmdd* is an optional report date. If no date is specified, prdaily produces a report of the current day's accounting information. Previous days' accounting reports can be displayed by using the *mmdd* option and specifying the exact report date desired.
- The `-l` option prints a report of exceptional usage by login name for the specified date. This option is used to determine which users are

consuming excessive amounts of system resources. The limits for exceptional usage are kept in the file `/usr/lib/acct/ptelus.awk` and can be edited to reflect your installation's requirements.

- Valid only for the current day's accounting, the `-c` option is used to get a report of exceptional resource usage by command. This option is used to determine which commands are using excessive amounts of system resources. The limits for exceptional usage are maintained in the file `/usr/lib/acct/ptecms.awk` and can be edited to reflect your system's needs.

The reports produced by runacct were described briefly in the previous subsection. Now the reports are discussed in more detail.

Daily line usage report

In the first part of this report, the `FROM/TO` banner should alert you to which period is being reported. The times are the date-time that the last report was generated by runacct, and the date-time that the current report was generated. It is followed by a log of system reboots, shutdowns, power failure recoveries, and any other records dumped into wtmp by the acctwtmp command.

The second part of the report is a breakdown of line utilization. The `TOTAL DURATION` shows how long the system was in a multiuser state. The columns of the report are defined as follows:

Column	Description
`LINE`	The terminal line or access port being reported on.
`MINUTES`	The total number of minutes that the line was in use during the accounting period.
`PERCENT`	The percentage of `TOTAL DURATION` that the line was in use: `PERCENT = (MINUTES / TOTAL DURATION)100`
`# SESS`	The number of times that this port was accessed for a login session.
`# ON`	Historically, this column displayed the number of times that the port was used to log a user on; but since `login` can no longer be executed explicitly to

log in a new user, this column should be identical to # SESS.

OFF — This column reflects not only the number of times a user logged off, but also any interrupts that occurred on the line. Interrupts occur on a port when getty is first invoked. A getty is invoked when the system is brought to run-level 2. This column comes into play when # OFF exceeds # ON by a large factor. This usually indicates that the multiplexer, modem, or cable is going bad, or that there is a bad connection somewhere. The most common cause of this is an unconnected cable dangling from the multiplexer.

During real time, wtmp should be monitored as this is the file from which connect session accounting is taken. If it grows rapidly, execute acctcon1 to determine which line is the noisiest. If the interrupting is occurring at a high rate, general system performance will be affected.

Daily resource usage report

This report gives a by-user breakdown of system resource usage. The format of this report is the same as that produced by the prtacct command.

Daily and monthly command cummary

These two reports are the same, except that the Daily Command Summary reports information only for commands executed since the last invocation of runacct; the Monthly Command Summary contains information on commands executed since the last invocation of monacct.

The output of this report is identical to that produced by acctcms. For definitions of the data found in this report, see the discussion of acctcms in the "Process accounting" section of this chapter.

Last login

This report simply shows the last date and time that each user logged in. The longer it has been since a particular user logged in, the more likely it is that the user's files could be archived, or maybe even that the user could be removed from the system.

Creating monthly accounting reports (monacct)

`monacct` creates monthly summary files and reports; the resulting output is stored in the directory `/usr/adm/acct/fiscal`. After creating its monthly reports, it removes the old daily accounting files from the directory `/usr/adm/acct/sum` and replaces them with new summary accounting files.

`monacct` should be invoked once each month or accounting period. Its syntax is

`monacct` *number*

where *number* indicates which month or period it is (01=January, 12=December). If *number* is not specified, monacct assumes that it is being invoked for the current month; this default is useful if monacct is executed via cron on the first day of each month (as described in the "Daily usage and installation" section of this chapter).

Descriptions of the files created in the `acct/fiscal` directory follow:

- `cms?` contains the total command summary file for the accounting period denoted by ?. The file is stored in internal summary format. Therefore, to display this file, you must use the acctcms command. The following example shows how to display this file for the month of June:

```
acctcms -a -s /usr/adm/acct/nite/fiscal/cms06
```

- `fiscrpt?` contains a report similar to that produced by prdaily. The report shows line and resource usage for the month represented by ?. The following would display the fiscal accounting file for the month of November:

```
cat /usr/adm/acct/nite/fiscal/fiscrpt11
```

- `tacct?` is the total accounting file for the month represented by ?. To display this file, you must use the `prtacct` command. The following

would display the total accounting summary file for the month of January:

```
prtacct/usr/adm/acct/fiscal/tacct01 "JANUARY TOTAL ACCOUNTING"
```

Updating the holidays file

The file `/usr/lib/acct/holidays` contains the information that System Accounting needs to distinguish between prime and non-prime time. It contains the following information:

Comment Lines

Comment lines are entered by placing an asterisk (*) as the first character in the line; they may appear anywhere in the file.

Year Designation Line

This line should be the first non-comment line in the file and must appear only once. The line consists of three four-digit numbers (leading blanks and tabs are ignored). The first number designates the year; the second denotes the time (in 24-hour format) that prime time starts; the third gives the time that prime time ends and non-prime time starts.

For example, to specify the year as 1994, prime time at 9:00 a.m., and non-prime time at 4:30 p.m., the following entry would be appropriate:

```
1994 0900 1630
```

A special condition allowed for in the time field is that 2400 is automatically converted to 0000.

Company Holiday Lines

These entries follow the year designation line. Company holidays are days when few people should be using the computer. Therefore, System Accounting assumes that non-prime time is in effect during the entire 24 hours of a specified holiday.

Company holiday lines have the following format:

day_of_year Month Day Description of Holiday

The *day_of_year* field is a number in the range 1 through 366, corresponding to the day of the year for the particular holiday (leading blanks and tabs are ignored). The remaining fields are simply commentary and are not used by other programs.

As delivered, the holidays file contains valid entries for prime/non-prime time, and holidays. You

should check this file and edit it as necessary to reflect your organization's requirements.

Fixing corrupted files

System Accounting files may become corrupted or lost. Some of these files can simply be ignored or restored from the files saved through backup procedures. However, certain files must be fixed in order to maintain the integrity of System Accounting. Two of the files that must be fixed are `/etc/wtmp` and `/usr/adm/acct/sum/tacct`.

Fixing wtmp errors

The `wtmp` files seem to cause the most problems in the daily operation of System Accounting. When the date is changed and SPP-UX is switched into multiuser mode, a set of date change records is written into `/etc/wtmp`. The `wtmpfix` command is designed to adjust the time stamps in the `wtmp` records when a date change is encountered. However, some combinations of date changes and reboots won't be caught by `wtmpfix` and cause `acctcon1` to fail. The following steps show how to fix a damaged `wtmp` file.

```
cd /usr/adm/acct/nite
```

```
fwtmp <wtmp.mmdd >wtmp.temp
```

Using an editor, delete corrupted records or delete all records from the beginning up to the date change

```
fwtmp -ic <wtmp.temt >wtmp.mmdd
```

```
rm wtmp.temp
```

If the `wtmp` file is corrupted beyond repair, create a null `wtmp` file. This will prevent any charging of connect time. `acctprc1` will not be able to determine which login owned a particular process, but it will be charged to the login that is first in the password file for that user ID.

Fixing tacct errors

If your installation is using System Accounting to charge users for system resource usage, the integrity

of sum/tacct is quite important. If sum/tacct ever becomes corrupted, then check the contents of sum/tacctprev with the command prtacct. If it looks correct, then the latest sum/tacct.*mmd* should be patched up, and sum/tacct should then be recreated. A simple patch procedure would be:

```
cd /usr/adm/acct/sum

acctmerg -a -v <tacct.mmdd >tacct.temp
```

Using an editor, delete corrupted records or delete all records from the beginning up to the date change

```
acctmerg -i <tacct.temp >tacct.mmdd

acctmerg tacctprev <tacct.mmdd >tacct

rm tacct.temp
```

Remember that monacct removes all the tacct.*mmdd* files; therefore, sum/tacct can be recreated by merging these files together.

Sample accounting shell scripts

grpdusg

This shell script displays disk space usage totals for the users who are members of a specified group. The syntax of this command is:

grpdusg *group_name*

where *group_name* is the name of the group for which disk space accounting information is to be generated.

For example,

```
grpdusg pseudo
```

generates disk space usage information for all the users in the group pseudo.

grpdusg shell script

```
# Check for the group-name parameter.
#
if      [ $# -ne 1 ]
then    echo "\nUsage: grpdusg group-name\n"
        exit 1
fi
echo    "\nOne moment please...\n"
#
# Use the find command to find all the files whose owners are members of
# group-name. Pipe the output from find into acctdusg which will accumulate
# disk space usage information for the users in group-name.
# NOTE:
# - accounting data is temporarily stored in _${1}_tmp
# - error messages are stored temporarily in _${1}_err #
- if files exist that have no owners, then the names of
# these files are stored in _no_owners
#
fn=_${1}_
find / -group $1 - hidden -print 2>${fn}err |acctdusg -u _no_owners >${fn}tmp
#
# Remove the _no_owners file if its size is not greater than zero.
#
if [ -s _no_owners ]
then    echo "\nFiles having no owners exist--check _no_owners\n"
else    rm _no_owners
        echo "\nAll files have owners-- _no_owners not created\n"
fi
#
# Use echo and awk to display disk usage totals for this group.
#
echo    "\nDisk space usage information (group is ${1}):\n"
awk 'BEGIN {print "\n_UIDUSER
NAME_________BLOCKS"}
        { sum += $3 ; # add up total disk blocks used
          print $0 # display information for user
        } END { print "\nTOTAL DISC SPACE USAGE= ", sum, "blocks" }' ${fn}tmp
#
# Remove temporary files, then exit.
#
rm ${fn}*
```

acct_bill

`acct_bill` takes as input a total accounting file and produces as output billing totals for all users found in the input file. The syntax of `acct_bill` is:

```
acct_bill [mmdd]
```

If the optional *mmdd* is not specified, then `acct_bill` takes as input the current day's total accounting file (`acct/nite/daytacct`); if *mmdd* is given, then input is taken from the total accounting file for the date specified by *mmdd* (`acct/sum/tacct`*mmdd*). Output is written to the file `bills`*mmdd*, where *mmdd* is the date given with the command, or the current date if *mmdd* was not specified with the command.

Examples

To generate billing information for the current day, enter:

```
acct_bill
```

and the billing information will be stored in the file `acct/sum/bills`*mmdd*, where *mmdd* is the current date.

To create billing information for January 23rd, you would enter:

```
acct_bill 0123
```

after which the billing information would be stored in the file named `acct/sum/bills0123`.

To automatically generate daily billing totals for all users, you should call `acct_bill` without the date argument from the `USEREXIT` state of `runacct`.

Output produced by acct_bill

The output of `acct_bill` contains one line per user and has the following format:

user_ID User_name billing_amount

where *user_ID* and *user_name* identify the user who is being billed, and *billing_amount* shows the total amount that the user is to be charged.

billing_amount is computed by multiplying accounting coefficients (found in the shell script) by columns of the report generated by `prtacct`. Assuming that billing amounts are in dollars, the coefficients (as they are shown in the shell script that follows) produce the following billing amounts:

- ten cents for every minute of prime CPU time consumed
- five cents for every minute of non-prime CPU time consumed
- a half cent for every prime kcore minute used
- two-tenths of a cent for every non-prime kcore minute
- a half cent for every prime connect time minute
- two-tenths of a cent for every non-prime connect minute
- two-and-a-half cents for every block of disk space used
- two-and-a-half cents for every process spawned by the user
- ten cents for every connect session
- each fee unit charged via `chargefee` counts as one cent

You should experiment with this command by altering the coefficients to see how *billing_amount* is affected. After gaining confidence with this shell script, you can alter the coefficients to suit your installation's needs.

acct_bill shell script

```
_date=`date +%m%d`
_outfile=/usr/adm/acct/sum/bills
_infile=/usr/adm/acct
#
# Set _infile and _outfile, based on whether or not mmdd was given
#
if      [ $# -eq 0 ]
then            # Generate billing data for current day.
        _infile=${_infile}/nite/daytacct
        _outfile=${_outfile}${_date}
else            # Create billing data for date given (mmdd).
        _infile=${_infile}/sum/tacct${1}
        _outfile=${_outfile}${1}
fi
#
# Create a file containing the ASCII equivalent of the input total
# accounting file (tacct_ASC.tmp_). The file can then be supplied as input
# to awk, which will generate billing data for each user.
# acctmerg -a -t &tacct_ASC.tmp # output TOTAL amount first
# acctmerg -a &tacct_ASC.tmp # append users' total accounting records
#
# Using awk, compute billing totals for each user in the total
# accounting file.
#
awk 'BEGIN {
        # ***************************************************
        # A C C O U N T I N G C O E F F I C I E N T S
        # ***************************************************
        cpu_P =0.10 # 0.10 monetary units per minute of prime CPU time
        cpu_NP=0.05 # 0.05 monetary units per non-prime CPU minute used
        kcm_P =0.005 # for prime kcore minutes consumed
        kcm_NP=0.002 # for non-prime kcore minutes used
        con_P =0.005 # prime connect (real) time
        con_NP=0.002 # non-prime connect time used
        blk = 0.025 # number of blocks used
        prc = 0.025 # number of processes spawned
        ses = 0.10 # number of connect sessions
        fee = 0.01 # 100 charge units per monetary unit
        # ***************************************************
        }
```

```
# Start computing billing amounts for each user.
        { _sum = cpu_P*$3 + kcm_P*$5 + con_P*$7 # compute prime usage
        _sum+= cpu_NP*$4+ kcm_NP*$6+ con_NP*$8 # add non-prime usage
        _sum+= blk*$9 + prc*$10 + ses*$11 + fee*$13 # add remaining amounts
        printf "%-8s %-10s %10.3f\n", $1, $2, _sum # display user total
        }' tacct_ASC.tmp_ >$_outfile # write output from awk to appropriate
file
rm tacct_ASC.tmp_ # remove the temporary ASCII file
```

System Accounting files

This section contains descriptions of the different files processed by SPP-UX System Accounting. The files are grouped according to the directory in which they are found.

Files in the /usr/adm directory

Filename	Contents
`diskdiag`	Diagnostic output from the execution of disk space accounting commands.
`dtmp`	Output from the `acctdusg` program.
`fee`	Output from the `fchargefee` command (ASCII total accounting records).
`pacct`	The current active process accounting file.
`pacct?`	Process accounting files switched via `turnacct` switch.

Files in the /usr/adm/acct/nite directory

Filename	Contents
`active`	Used by `runacct` to record progress. It contains warning and error messages. active *mmdd* is the same as active after `runacct` detects an error.

`ctacct.` *mmdd*	Total accounting records created from connect session accounting where *mmdd* is the month and day the file was created.
`ctmp`	Output of `acctcon1`; connect session records.
`daycms`	ASCII daily command summary used by `prdaily`.
`daytacct`	Total accounting records for current day.
`disktacct`	Total accounting records created by the `dodisk` command.
`fd2log`	Diagnostic output from the execution of `runacct` (refer to `crontab` entry).
`lastdate`	The last day that runacct was executed, in `date | #` *+%m%d* format. See the date(1) man page for a description of *+%m%d* date format.)
`lock` and `lock1`	Used to control serial use of `runacct`.
`lineuse`	Terminal (tty) line usage report used by `prdaily`.
`log`	Diagnostics output from `acctcon1`.
`log` *mmdd*	Same as log after `runacct` detects an error.
`reboots`	Contains beginning and ending dates from `wtmp`, and a listing of reboots.
`statefile`	Used to record the current state being executed by `runacct`.
`tmpwtmp`	`wtmp` file, corrected by `wtmpfix`.
`wtmperror`	Error messages, if any, from `wtmpfix`.
`wtmperror` *mmdd*	Same as wtmperror after `runacct` detects an error.
`wtmp.` *mmdd*	The previous day's `wtmp` file.

Files in the /usr/adm/acct/sum directory

Filename	Contents
`cms`	Total command summary file for current month in internal summary format.
`cmsprev`	Command summary file without latest update.
`daycms`	Command summary file for previous day in internal summary format.
`loginlog`	Shows the last login date for each user.
`rpt` *mmdd*	Daily accounting report for date *mmdd*.

`tacct`	Cumulative total accounting file for current month.
`tacctprev`	Same as `tacct` without latest update.
`tacct` *mmdd*	Total accounting file for date *mmdd*.
`wtmp.` *mmdd*	Saved copy of `wtmp` file for date *mmdd*. Removed after reboot.

Files in the /usr/adm/acct/fiscal directory

Filename	Contents
`cms?`	Total command summary for month ? in internal summary format.
`fscrpt?`	Report similar to prdaily for the month ?.
`tacct?`	Total accounting file for the month ?.

SPP-UX system tunable parameters

A

This appendix describes the SPP-UX system tunable parameters.

SPP-UX system tunables file

The file `/os/tunables` contains tunable parameters that allow the system administrator to adjust system performance. The parameters from the `/os/tunables` file are read and set for the system each time SPP-UX boots. If you change the values for parameters in `/os/tunables`, the new values will take effect the next time the system is booted.

The `/os/tunables` file contains parameters for both the SPP-UX microkernel and the SPP-UX server.

When you install a new version of SPP-UX, your `/os/tunables` file is not automatically updated to add new tunables. Instead, a new tunables file is installed in the `/etc/newconfig` directory. After installing a new version of SPP-UX, check `/etc/newconfig/tunables` against your `/os/tunables` file to see if new tunables have been added or if the range of values for tunables has changed.

Table 3 lists the microkernel tunable parameters:

Table 3 Microkernel tunable parameters

parameter	range of values/ defalut value	description
Crashdump, do_crashreport	*0 | 1* 1	Determines whether a crashdump report is generated when the system crashes
Crashdump, do_kerneldump	*0 | 1* 1	Determines whether a crashdump of the microkernel is generated when the system crashes
Crashdump, do_swdump	*0 | 1* 1	Enables (1) or disables (0) the crashdump capability
Crashdump, panic_graceful	*0 | 1* 1	Determines whether messages are printed when a microkernel panic occurs
Event Logger, buffer size	*0-65536* 0	Size of the event logger buffer in bytes; larger sizes improve event logging at the expense of available physical memory
LCD Heart Beat, control	*0 | 1* 1	Sets (1) or clears (0) a flag that determines whether the Exemplar LCD displays a heartbeat and the state of each CPU, updated every 1/4 second

Table 3 lists the server tunable parameters:

Table 4 Server tunable parameters

parameter	range of values/ default value	description
Fileserver, buffer_cache_percent	*0-100* 10	The buffer cache for each node to the specified percent of physical memory
Fileserver, disk_wdb_size	*0-128*1024* 1024	The maximum number of wired device buffers for hard disk drives
Fileserver, tape_wdb_size	*0-16*1024*1024* 128*1024	The maximum number of wired device buffers for raw tape devices
Server, distribute_panic	*0 | 1* 1	Sets (1) or clears (0) a flag that determines whether a panic in the node's file server is distributed to all other nodes' file servers, or whether a panic in the node's file server causes only the local node's server to shut down
Server, acctresume	*0-100* 4	The percentage of file system space that must be free in order to reactivate process accounting after it is suspended due to insufficient free space
Server, acctsuspend	*0-100* 2	The percentage of file system space that must be free in order to allow process accounting
Server, dfldsiz	*512*1024-* *0x0c0000000* 64*1024*1024	The default size, in pages, of a process's data segment
Server, dflssiz	*512*1024-* *0x0c0001000* 64*1024*1024	The default size, in pages, of a process's stack
Server, dst	*0 | 1* 1	Sets (1) or clears (0) a flag that specifies whether daylight saving time would be used

Table 4 Server tunable parameters (continued)

parameter	range of values/ default value	description
Server, incksum	*1 | 2 | 3* *2*	Internet Protocol (IP) checksumming method. This parameter should be set to a value of 2 under normal circumstances. If the `netstat -s` command shows an abnormally high number of `ip`, `tcp`, or `udp` checksum errors, contact the Convex TAC for assistance in changing the value of this parameter.
Server, maxdsiz	*512*1024-* *0x0c0000000* 512*1024*1024	The maximum size, in pages, of a process's data segment
Server, maxfiles	*1-512* 256	The maximum number of files a process can have open at one time. This limit can be changed by using `setrlimit(2)`
Server, maxssiz	*512*1024-* *0x0c0001000* 512*1024*1024	The maximum size, in pages, of a process's stack
Server, maxuprc	*8-1024* 256	The maximum number of processes a user can have
Server, msgmax	*0-65536* 8192	The maximum number of processes a user can have
Server, msgmnb	*0-65536* 16384	The maximum number of bytes allowed for all queued messages
Server, msgmni	*1-int_max* 50	The number of message queue identifiers
Server, msgtql	*1-int_max* 40	The number of message queue headers
Server, nbuf	*1-80* 10	The number of file system buffer cache headers

Table 4 Server tunable parameters (continued)

parameter	range of values/ default value	description
Server, ncallout	*26-int_max* nproc+16	The number of timeouts that can be pending simultaneously
Server, nfile	*32-int_max* (16*(nproc+16+ maxusers)/10+32)	The maximum number of open files
Server, nflocks	*50-400* 200	The maximum number of file locks
Server, nmount	*1-int_max* 20	The maximum number of mounted file systems
Server, nproc	*10-int_max* (20+8*maxusers)	The maximum number of processes that can exist at one time
Server, npty	*16-27900* 60	The maximum number pseudo-terminals that can exist at one time
Server, semaem	*0-int_max* 16384	The maximum value by which a semaphore can be adjusted due to the death of a process
Server, semmni	*2-int_max* 64	The number of semaphore identifiers
Server, semmns	*2-int_max* 128	The maximum number of semaphores
Server, semmnu	*1-int_max* nproc_d	The maximum number of processes that can have `semaphore undo` requests on a semaphore
Server, semume	*1-int_max* 10	The maximum number of semaphores on which a process can have can have a pending `semaphore undo` request
Server, semvmx	*1-65535* 32767	The maximum value of a semaphore

Table 4 Server tunable parameters (continued)

parameter	range of values/ default value	description
Server, shmmax	*2048-0xC00000000* 0x4000000	The maximum number of bytes in a shared memory segment
Server, shmmni	*3-1024* 200	The maximum number of shared memory segments
Server, shmseg	*3-1024* 120	The maximum number of shared memory segments that can be attached to a process at one time

Table 5 lists HP-UX tunable parameters that are not implemented in SPP-UX due to architecture differences:

Table 5 HP-UX tunables not implemented in SPP-UX

bootspinlocks	netisr_priority
bufpages	ngcsp
check_alive_period	nstlbe
dmmax	nswapdev
dmmin	nswapfs
dmshm	ntext
dmtext	num_codes
dskless_cbufs	retry_alive_period
dskless_fbufs	scroll_lines
dskless_mbufs	selftest_period
dskless_node	semmap
maxswapchunks	server_node
maxtsiz	serving_array_size
minswapchunks	timeslice
msgmap	unlockable_mem
msgseg	using_array_size

Crashdump

B

This appendix describes the SPP-UX `crashdump` utility.

Oveview

The SPP-UX `crashdump` utility stores information about the state of the system to a raw disk partition in the event of a system crash. This information can be useful to Convex in some cases to help determine the cause of the system crash.

The `crashdump` utility is part of the SPP-UX microkernel. It runs when SPP-UX terminates abnormally. `crashdump` writes data to a raw disk partition that you create for this purpose.

When a system crash occurs for a reason you do not understand, contact the Convex Technical Assistance Center (TAC). If the TAC determines that you should send a crashdump file to Convex, create a crashdump file using the `crashutil` command, make a tape containing the crashdump file, and mail it to Convex. See "Technical assistance" on page xvii for information about contacting the Convex TAC.

Creating a crashdump partition

You must create a crashdump partition on a disk drive on your Exemplar system to hold the data generated by `crashdump`. Use the `diskutil` command to create, modify, or delete a crashdump partition. (For more information about `diskutil` and the `diskutil` commands, see "The diskutil disk utility" on page 125.) Use the following procedure to make a crashdump partition:

1. Enter the `diskutil` command, followed by the `show disks` and `show partitions` commands, in order to see the available disks and partitions on your system:

```
diskutil

show disks

show partitions
```

2. Select the disk on which you wish to make a crashdump partition with the `select disk` command. Use the `make partition` command to create a crashdump partition, then use the `set partition` command to set the `crashdump` flag for this partition:

```
select disk sd0

make partition c size 100M after b description "crash"

set partition c flag crashdump
```

Instead of creating a new crashdump partition, you can change an existing partition to a crashdump partition. First make sure that all useful data has been removed from the partition, since it will be overwritten by `crashdump`. Also be sure that the selected partition is not mounted. Then use the `set partition` command to identify this partition ad a crashdump partition:

```
set partition c flag crashdump
```

To stop using a partition as a crashdump partition, use the `unset partition` command:

```
unset partition sd0c flag crashdump
```

It is possible to create more than one crashdump partition; however, `crashdump` will use only the first crashdump partition it finds.

Creating a crashdump file(crashutil)

If the Convex TAC has determined that you should send a crashdump to Convex, use the crashutil command to create a crashdump report. The `crashutil` command has the following format:

`crashutil` *block-partition outfile*

block-partition is the name of the raw disk partition containing the crashdump data. *outfile* is the output file to which the crashdump file is to be written. For example, if your crashdump partition is `/dev/dsk/sd0c`, and you wish to write the crashdump file to `/tmp/crashfile`, you would enter:

```
crashutil /dev/dsk/sd0c /tmp/crashfile
```

The Convex TAC will give you instructions on further handling of the crashdump file.

Only one set of crashdump data resides in a raw crashdump partition at any time. When a crashdump takes place, it overwrites any previously written crashdump data. It is important to run `crashutil` as soon as possible after recovering from a crash in order to prevent crashdump data from being lost.

Index

Symbols

.cshrc file 20
.exrc file 23
.kshrc file 20
.login file 20
.profile file 20

A

access control lists 61
access permissions 53
accessing, the system console 2
acctresume, server tunable 259
acctsuspend, server tunable 259
ACL. See access control list
adding
  network-based printer 153, 169
  remote printer 150, 164
adding a group using SPP-UX commands 98
Adding a user 78
adding a user group 74
adding users to groups using SPP-UX commands 104
ARPA Services 17

B

bcheckrc file 45
bchecrc command 45
block device 117
block special file 117
boot sequence 44
boot-time paramaters, *See* tunables
brc file 45
buffer size, microkernel tunable 258
buffer_cache_percent, server tunable 259

C

cancel model script (printer) 133
catman command 15
caution
  explained xv, 13, 46, 79, 93, 114
CD-ROM file system 117, 122
chacl command 61
changing a user's primary group using SPP-UX commands 103
character special file 117
chgrp command 61
chmod command 60
chown command 60
coherent toroidal interconnect cache 27
command
  accept 135, 173, 181
  cancel 178
  disable 135, 171, 175, 179
  enable 135, 175, 180
  lpadmin 164, 165, 168, 170
  lpalt 181, 185
  lpana 177, 186, 187
  lpmove 180
  lpsched 140, 156, 177, 180
  lpshut 164, 172, 177
  lpstat 167, 177, 178, 182
  reject 135, 173, 179
  rlpdaemon 142
complex configuration file
  Subcomplex Manager
    complex configuration file 29
console file 45
contact command xvii
controlling
  access to the system 55
  file access 55, 59
  groups 55
  run-levels 55, 62
  user accounts 55
crashdump tunables, do_crashreport 258
crashdump tunables, do_kerneldump 258
crashdump tunables, panic_graceful 258
creating
  a printer class using lpadmin 169
Creating a new user account 6
csh.login file 20
customizing
  printer model scripts 139
Customizing the system 11
customizing the system 19
cylinder 118

D

deactivating a user's account 71
Deactivating a user's account, using SPP-UX commands 92
default group 54
default printer 134
default tunables
  microkernel, table 258

server, table 259– 262
device drivers
printers 163
device file 118
device files 140
device swap space 118
dfldsiz, server tunable 259
dflssiz, server tunable 259
disabling printers 159, 175
disk layout 122
disk quotas 118
disk section 118
disk utilities 125
disk_wdb_size, server tunable 259
disktab file 122
diskutil
Exit command 126
Help command 126
MAKe Partition command 127
MAP command 127
Prepare Disk command 127
SELect Disk command 128
SELect Volume command 128
SET Partition command 128
SHow Directory command 128
SHow Partitions command 128
UNMap Disk command 128
UNSet Partition command 129
diskutil command 125
DISPLAY environment variable 5
displaying/assigning special group privileges using SPP-UX commands 108
displaying/modifying a user's account information using SPP-UX commands 95
distribute_panic, server tunable 259
do_crashreport microkernel tunable 258
do_kerneldump microkernel tunable 258
dst, server tunable 259

E

editing environment 23
effective group 53
electronic mail
setting up 11, 16
electronic news
setting up 11
elm mail program 17
enabling printers 160, 175
environment variable 138
LPDEST 138
event logger tunables, buffer size 258
EXINIT environment variable 23
exrc file 20

F

file
/os/tunables 257
/usr/spool/lp/cmodel/rcmodel 142, 165
/usr/spool/lp/lpana.log 177, 186
/usr/spool/lp/smodel/rsmodel 142, 165
file access 59
permissions 60
file system 118, 120
types 121
file system utilities 125
file systems
mounting 11
fileserver tunable, buffer_cache_percent 259
fileserver tunable, disk_wdb_size 259
fileserver tunable, tape_wdb_sizeSPP-UX
fileserver tunable, tape_wdb_size 259
fragment 118
fsclean command 45, 46

G

getting help in SAM 8
group 53
group file 19
group ownership 53
group passwords 58
group_ID 53
groups
primary 58
special 58

H

HFS file system 118, 121
hosts file 21

I

incksum, server tunable 260
init file 45
inittab file 19, 21, 45
inode 119
installing
optional software 10
installing and upgrading
optional software 9
SPP-UX 9
installing and upgrading optional software 9
interface scripts (printer) 132, 135
interfaces, SAM 4
issue file 20, 21

K

kernel 119

L

LCD heart beat microkernel tunable 258
line printer spooling system 131
line-printer spooler
  setting up 17
local printer 133, 141
log in 54
logical printer 133
login accounts
  creating 11
login environment 22
login prompt 21
long file names 119
LPDEST environment variable 138
lpss
  accepting print requests 145, 173
  adding a network-based printer 153, 169
  adding a remote printer 150, 164
  adding printers 167
  canceling print requests 178
  changing a printer fence 176
  changing a printer fence priority 161
  checking status 158, 177
  collecting printer activity statistics 146, 177, 186
  commands 135
  components of 136
  controlling with SAM 147, 149, 162
  controlling with SPP-UX commands 147, 162, 187
  creating a printer class 169
  device files 164
  disabling printers 145, 159, 175
  displaying printer activity statistics 186, 187
  enabling printers 145, 160, 175
  interface scripts 135
  LPDEST environment variable 138
  moving all print requests 179
  moving selected print requests 181
  network-based printer 141
  plotters 144
  print destinations 137
  print request 133
  print request identification number 141
  print request priorities 143, 185
  print requests 135
  printer classes 137
  printer interface scripts 138
  printer models 139, 164
  printer names 136, 164
  printer priorities 143, 165
  printer queues 135
  priorities 143
  rejecting print requests 145, 173
  remote print requests 142
  remote spooling 135, 142
  removing a printer 155
  removing a printer class 171
  request directories 135
  scheduler 140
  setting up 147, 162
  starting a scheduler 177
  starting scheduler 156, 172
  stopping scheduler 156, 172, 177
  system default printer 138, 150, 164
  viewing printer request status 149, 182
  viewing printer status 149, 182

M

mailx mail program 17
MAKe Partition command, diskutil 127
man pages 10
  preformatting 10
  setting up 15
managing run-levels 111
MAP command, diskutil 127
maxdsiz, server tunable 260
maxfiles, server tunable 260
maxssiz, server tunable 260
maxuprc, server tunable 260
message of the day 22
model script (printer) 133
motd file 20, 22
mount 119
mount directory 119
mount point 119
mountable file system 120
msgmax, server tunable 260
msgmnb, server tunable 260
msgmni, server tunable 260
msgtql, server tunable 260
multi-user mode 54

N

nbuf, server tunable 260
ncallout, server tunable 261
network-based plotter 133, 141
network-based printer 133, 141, 153, 169
networking
  setting up 10
news
  electronic 11
  program 18
  setting up 18
nfile, server tunable 261
nflocks, server tunable 261
NFS client 119

NFS file system 119, 121
NFS server 119
nmount, server tunable 261
NONHPTERM file 23
Notational conventions xiv
notational conventions xiv
note
  explained xv, 6, 49, 58, 68, 79, 110, 199
nproc, server tunable 261
npty, server tunable 261

O

optional software
  installing 10
Ordering documents xvi
ownership 54

P

panic_graceful microkernel tunable 258
passwd file 19
peripherals
  adding a network-based printer 153, 169
  adding a remote printer 150, 164
  removing a printer 171
plotter 144
post-installation tasks 10
  first time 11
power failure 50
Prepare Disk command, diskutil 127
primary group 54
primary groups 58
print destinations 132, 137
print priority 134
print queues 134
print request 133
print request identification number 133, 141, 178
print requests 135, 141, 178
Printer
  classes 137
printer
  cancel model script 133
  classes 169
  device drivers 163
  environment variable LPDEST 138
  interface scripts 132, 138
  local 133
  logical 133
  model scripts 133, 139
  names 136
  network-based 133
  queues 135
  removing using SAM 155
  removing using SPP-UX commands 171
  statistics 146, 177, 186
  status model script 133
  system default 138
printer class 133, 171
printer fence 161, 176
printer interface scripts 135
printer name 134
printer spooler
  setting up 11
priorities (printers and print requests) 143
profile file 20

R

rc file 19, 45
  editing 21
rc.local file 21
reactivating a user's account 72
Reactivating a user's account using SPP-UX commands 94
remote printer 134, 141, 143, 150, 164
remote spooling 134
remote Spooling Daemon 142
remote spooling daemon 134
removing
  printer class 171
  printers 155, 171
removing a group using SPP-UX commands 101
Removing a user group 76
Removing a user, using SPP-UX commands 88
removing users from groups using SPP-UX commands 106
request directories 135
root directory 119
root password 10
  setting 14
run-level 54
run-levels 111
  controlling 62

S

SAM 3
  adding a group 74
  adding users 65
  Creating a new user account 6
  customizing 68
  deactivating a user's account 71
  Entering 148
  exiting 148
  getting help 8
  halting the system 50
  quitting 5
  reactivating a user's account 72
  rebooting the system 50
  removing a user 67
  Removing a user group 76

sample session 6
starting 4
using 4
SAM, Interfaces 4
SCM 3, 27– 42
See also CTI cache 27
See also SCM
See also Subcomplex Manager
SELect Disk command, diskutil 128
SELect Volume command
diskutil 128
semaem, server tunable 261
semmni, server tunable 261
semmns, server tunable 261
semmnu, server tunable 261
semume, server tunable 261
semvmx, server tunable 261
server set 27
server tunables
acctresume 259
acctsuspend 259
dfldsiz 259
dflssiz 259
distribute_panic 259
dst 259
incksum 260
maxdsiz 260
maxfiles 260
maxssiz 260
maxuprc 260
msgmax 260
msgmnb 260
msgmni 260
msgtql 260
nbuf 260
ncallout 261
nfile 261
nflocks 261
nmount 261
nproc 261
npty 261
semaem 261
semmni 261
semmns 261
semmnu 261
semume 261
semvmx 261
shmmax 262
shmmni 262
shmseg 262
SET Partition command, diskutil 128
set_parms command 10
Setting up an SPP-UX system 1
shmmax, server tunable 262
shmmni, server tunable 262
shmseg, server tunable 262
SHow Directory command, diskutil 128
SHow Partitions command, diskutil 128
shutdown command 47
shutting down the system 46
single-user mode 44, 47, 54
sn.cnsld daemon 1
sn_cnsl command 2
special group privileges 108
special groups 58
spooler
setting up 17
SPP-UX
adjusting tunables 257, 265
changing tunables 257, 265
crashdump tunables, do_crashreport 258
crashdump tunables, do_kerneldump 258
crashdump tunables, do_swdump 258
crashdump tunables, panic_graceful 258
event logger tunables, buffer size 258
fileserver tunable, buffer_cache_percent 259
fileserver tunable, disk_wdb_size 259
installing and upgrading 9
microkernel tunable 257
server tunables, table 259– 262
starting 44
tunables, system 257, 265
SPP-UX configuration
printer drivers 163
SPP-UX, setting up 1
SPP-UX, system console 1
Starting SAM 4
starting SPP-UX 44
status model script 133
subcomplex
subcomplex
defined 27
Subcomplex Manager 27– 42
graphical user interface 28– 31
interfaces 28
main window 31
system reconfiguration 30
viewing system resources 29
windows 31– 37
system accounting
setting up 11, 19
system administration run-level 114, 115
system administrator responsibilities
adjusting system performance with tunables 257
setting tunable values 257
system administrator's responsibilities 3
system console 1
system console, accessing 2
System default printer 138
system default printer 134, 150, 164
system files 122
system startup
customizing 20
system subcomplex 27, 28

T

TAC xvii
tape_wdb_size, server tunable 259
Technical assistance xvii
TERM environment variable 24
terminal types 23
terminfo file 23, 24
ttytype file 19
tunable paramaters, *See* tunables
tunables
  adjusting 257, 265
  changing 257
  crashdump 258
  default microkernel value, table 258
  default server values, table 259– 262
  event logger 258
  fileserver 259
  HP-UX, unsupported 263
  LCD heart beat 258
  microkernel, table 258
  range of values 258– 262
  server, table 259– 262
  setting 257
  SPP-UX 257– 262
  unsupported HP-UX 263
  values 258– 262
typographic conventions xiv

U

UNMap Disk command, diskutil 128
UNSet Partition command, diskutil 129
update.log file 10
user account 54
user accounts 63
user groups
  creating 11
user_ID 54

ORDER NUMBER
DSW-853

DOCUMENT NUMBER
710-029330-000