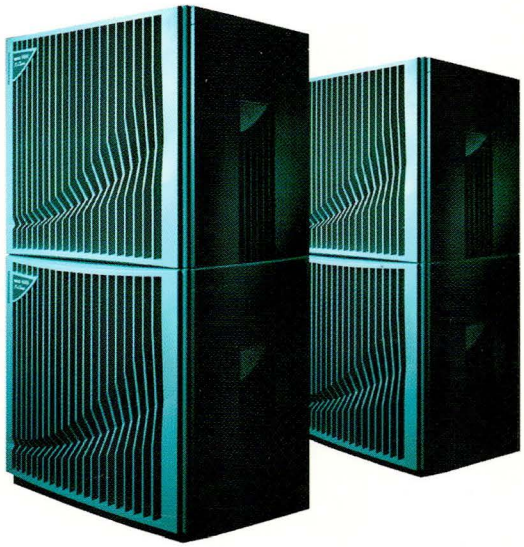
HEWLETT® PACKARD

Exemplar
S-Class and
X-Class Servers



SPP-UX
System Administration Guide

Fifth Edition

SPP-UX System Administration Guide

Exemplar S-Class and X-Class Servers

B5655-90023

Fifth Edition

June 1997

Hewlett-Packard Company
Convex Division
Richardson, Texas
United States of America

SPP-UX System Administration Guide

Exemplar S-Class and X-Class Servers

B5655-90023

Notice

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

This entire book is recyclable.

Printed in the United States of America

Revision Information for

SPP-UX System Administration Guide

Exemplar S-Class and X-Class Servers

Edition	Document No.	Description
Fifth	B5655-90023	Released June 1997 with SPP-UX V5.2.
Fourth	B5655-90002	Released January 1997.
Third	710-029330-002	Released June 1996.
Second	710-029330-001	Released July 1995.
First	710-029330-000	Initial release October 1994.

Contents

Figures

Tables

Preface

This document describes the features of SPP-UX, the operating system for Exemplar S-Class and X-Class Technical Servers.

Material in this manual assumes that the reader has a basic knowledge of SPP-UX or HP-UX.

Notational conventions

This section discusses notational conventions used in this book.

`Bold monospace`

In command examples, text shown in **`bold monospace`** identifies user input that must be typed exactly as shown.

`Monospace`

In paragraph text, `monospace` identifies command names, system calls, and data structures and types.

In command examples, `monospace` identifies command output, including error messages.

In command syntax diagrams, text shown in `monospace` must be typed exactly as shown.

Italic

In paragraph text, *italic* identifies new and important terms and titles of documents.

In command syntax diagrams, *italic* identifies variables that must be supplied by the user.

Bold

The **bold** character format is used for special emphasis.

{ }

In command syntax diagrams, text surrounded by curly brackets indicates a choice. The choices available are shown inside the curly brackets and separated by the pipe (|) sign.

The following command example indicates that you can enter either a or b:

```
command {a | b}
```

[ ]

In command syntax diagrams, square brackets indicate optional data.

The following command example indicates that the variable *output_file* is optional:

`command` *input_file* [*output_file*]

...

In command syntax, horizontal ellipsis shows repetition of the preceding item(s).

The following command example indicates you can optionally specify more than one *input_file* on the command line:

`command` *input_file* [*input_file ...*]

KEYCAP

In paragraph text, text shown in **KEYCAP** indicates keyboard keys you must press to execute the command. For example, **RETURN** refers to the carriage return key.

Two **KEYCAP** terms separated by a hyphen indicate two keys that you must press simultaneously. For example, **CTRL-d** indicates that you must press the **d** key while holding down the **CTRL** key.

Associated documents

For more information on Exemplar S-Class and X-Class Technical Servers, you can order these books from Hewlett-Packard:

- *NQS System Administration Guide: Exemplar S-Class Servers* (B5589-90008) Describes basic NQS concepts, configuration, and maintenance.
- *OpenBoot Quick Reference: Exemplar S-Class and X-Class Servers* (B5655-90028) Standard reference for OpenBoot PROM.
- *Exemplar Networking Guide: S-Class Servers* (B5655-90004) Describes tasks required to configure and administer Exemplar S-Class Server associated networking hardware and software based on the type of network at your site.
- *Exemplar Networking Guide: X-Class Servers* (B5655-90029) Describes tasks required to configure and administer Exemplar X-Class Server networking hardware and software based on the type of network at your site.

- *Exemplar Architecture: S-Class and X-Class Servers* (A4716-90001) Provides technical information and optimization applications for Exemplar S-Class and X-Class architecture.
- *FlexLM End User Manual* (B5655-90010)Provides a licensing overview for system administrators and end users. Topics covered include: the license file and license administration tools.
- *HP-UX 10.0 File System Layout Whitepaper* (March 1995) Explains the reasoning for and impacts of the new file system layout and system startup/shutdown model. An ASCII file is located on your Exemplar Technical Server in /usr/share/doc/filesys.txt.
- *Managing software with SD-UX* (B2355-90080) Introduces Software Distributor (SD). Covers distributing and administering software, as well as creating software packages with SD.

Ordering documents

To order additional copies of this document or other documents listed in 'Associated Documents,' send requests to:

Hewlett-Packard Company
Convex Division
Customer Service
P.O. Box 833851
Richardson TX 75083-3851 USA

Please include the order number (xxxxx-9xxxx number) or the exact title of the document.

Technical assistance

If you have questions that are not answered in this book, contact the Hewlett-Packard 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 your local Hewlett-Packard office.

You can also use the `contact` utility, if you would like to report any problems you may have with SPP-UX or its associated documentation.

Starting and stopping SPP-UX

1

This chapter provides information about starting and stopping SPP-UX. Tasks covered include:

- Starting your Exemplar S-Class and X-Class Technical Servers
- Using the console
- Customizing startup and shutdown processes
- Starting SPP-UX
- Setting run levels
- Stopping SPP-UX

Terms

Terms used in this chapter include:

OBP command prompt

Appears in a console window after the Exemplar Technical Server powers on. The prompt looks like:

```
(0:1) ok
```

where

`0`

Is the node number.

`1`

Is the CPU number.

`ok`

Is the command prompt.

Open Boot PROM (OBP)

Tests the Technical Server's hardware, provides a complete and accurate description of all available hardware, and boots the operating system. Resides on the Exemplar Technical Server. See the *OBP Quick Reference: Exemplar S-Class and X-Class Technical Servers* for more information about OBP.

dynamic files/directories

Private or host specific. Examples include log or configuration files, temporary files, and user directories.

hypernode

Set of processors and physical memory organized as a symmetric multiprocessor (SMP) running a single image of the operating system. An Exemplar Technical Server consists of one or more nodes, with a high-speed ring interconnect.

multiuser mode

Allows multiple users to access the system simultaneously. See also single-user mode.

run level

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

run-level s

Restricts user input to the console. Also known as single-user mode.

single-user mode

Restricts user input to the console. Also known as run-level s, system administrators use this mode to prevent users from accessing the system during system maintenance.

static files/directories

Sharable by other systems. Examples include executable files, libraries, and system startup directories.

Changes specific to this release

The startup/shutdown model allows you to set and modify system behavior by changing script variables. Non-modifiable execution scripts evaluate the variables and start or stop subsystems according to the arguments you supply. This model separates script execution and variables into the following locations:

Execution scripts

Located in /sbin/init.d. These scripts start up and shut down every subsystem. Execution scripts are not modifiable and will be over-written in subsequent releases of SPP-UX.

Configuration variable scripts

Contained in /etc/rc.config.d. These scripts control system behavior.

Link files

Located in /sbin/rc*.d. These links control the sequencing order of the execution scripts. Do not modify the ordering of these link files.

For more information about the new startup and shutdown processes see "Customizing startup and shutdown processes" on page 9.

If you have modified rc scripts to include customized functionality, move the script information to comply with the new startup/shutdown model. For more information, see the *HP-UX 10.0 File System Layout Whitepaper*, located on your system in /usr/share/doc/filesys.txt.

In the new model, the /etc directory can contain only configuration data. In most cases, executable files and scripts previously located in /etc are now in /usr/sbin and transitional links are provided to ease conversion; for instance, /etc/reboot is a link to /usr/sbin/reboot. However, in single-user mode, /usr may not be mounted. In this case, use the version of the program in /sbin.

The /etc/shutdown.d directory is not supported. If you have scripts in /etc/shutdown.d, migrate them to the new method. See "Customizing startup and shutdown processes" on page 9 for an overview and an example of the new method.

For more information about file system layout, see "Managing SPP-UX file systems," on page 55.

Starting your Exemplar S-Class and X-Class Technical Server

Bringing your Exemplar Technical Server to a usable state involves two systems and their hardware and software. This section provides a brief overview of the process; for complete instructions see the *Exemplar Distribution Notice* that accompanies your system.

An Exemplar Technical Server consists of two systems:

- Exemplar test station
  - Boots the Exemplar Technical Server
  - Monitors the Exemplar Technical Server for hardware errors
  - Debugs a hung system
  - Runs HP-UX
- Exemplar Technical Server
  - Hosts OpenBoot PROM (OBP) software
  - Consists of one or more hypernodes
  - Runs SPP-UX

The boot procedure differs according to the value of the OBP NVRAM parameter `auto-boot?` After you power-up your Exemplar Technical Server, if `auto-boot?` is set to:

- `True`, OBP automatically starts SPP-UX. (Press the **ESC** key to interrupt the boot process and enter OBP commands.)
- `False`, you must:
  - Start OBP at the test stations's HP-UX prompt by entering the following command:

    `do_reset`
  - Start the Exemplar Technical Server using default values at OBP's default prompt. Enter:

    `boot`

Figure 1 illustrates the initialization and start-up process.

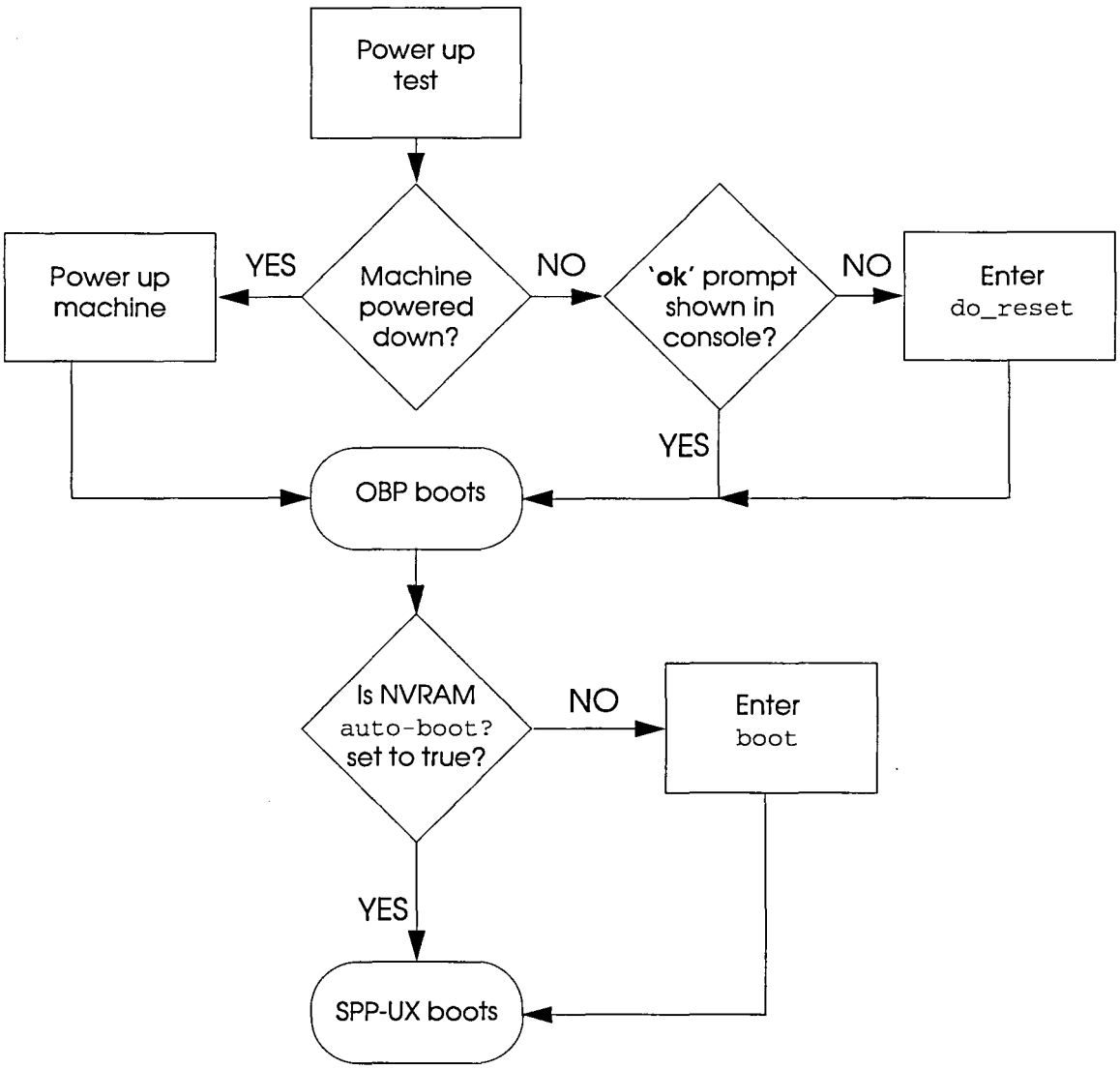


Figure 1 SPP-UX boot process

See the *OBP Quick Reference: Exemplar S-Class and X-Class Technical Servers* for more information.

Using the console

The console serves as a communication device between you and your Exemplar Technical Server. Virtual consoles are also used to monitor specific operations, like a system software crash dump.

This section familiarizes you with the console and shows you how to:

- Start the console
- Access the console remotely
- Access system logs

Starting the console

The SPP-UX console window's title is System Console.

The console server program automatically starts the console on the test station when SPP-UX boots. If the console dies or stops running, restart it by completing the following steps:

Step 1 Log onto the test station as root.

Step 2 Start the console. Enter:

`/spp/scripts/sppconsole` *hostname*

where *hostname* is the machine name. The `console` program polls *hostname* as its primary server.

For more information see the console(8), conserver(8), and conserver.cf(5) man pages, available on the test station.

Accessing the console remotely

Use `/spp/etc/console` to access the console from a system other than the test station. This command allows you to watch or to assume control of the console window.

Getting help for the `console` command

The `/spp/etc/console` command's `-h` option prints help for each of its options, as shown in the following output:

```
console -h

console: usage [-aAfFsS] [-v] [-e esc] [-M mach] [-P port] machine
console: usage [-hpuVw]
console: usage [-qQ] [-v] [-M mach]
a(A)    attach politely (and replay last 20 lines)
d(D)    display (local) daemon version
e esc   set the initial escape characters
f(F)    force read/write connection (and replay)
h       output this message
M mach  master server to poll first
q(Q)    send a quit command to the (local) server
r       connect to the console group only
```

```
s(S)    spy on a console (and replay)
u       show users on the various consoles
v       be more verbose
V       show version information
w       show who is on which console
x       examine ports and baud rates
```

Watching the console remotely

Any root user can watch the console via a remote login to the test station, so it is possible to have many different processes watching the console at the same time. Only one window can actually control the console, see "Controlling the console remotely" on page 7 for more information.

To monitor the console from a system other than the test station, complete the following steps:

Step 1 Remotely log in to the test station as root.

Step 2 Watch the console with the following command:

`/spp/etc/console -{s | S}` *hostname*

where:

`-s`

Watches the main SPP-UX console; all messages written to the console are duplicated on your screen.

`-S`

Watches the main SPP-UX console and prints the last message stored in the message buffer; all messages written to the console are duplicated on your screen.

hostname

Identifies the current host's console.

Step 3 Exit the session with the following command:

CTRL-Ec.

The period is part of the command.

Controlling the console remotely

You can perform system maintenance or diagnostics remotely by assuming control of the console from a remote terminal. Upon gaining control of the console, you have write access to that window.

Only one window can be active at a time.

To assume control of the console, complete the following steps:

Step 1 Remotely log in to the test station as root.

Step 2 Take control of the console with the following command:

`/spp/etc/console -{f | F}` *hostname*

where:

`-f`

Forces control of the SPP-UX console to your screen.

`-F`

Forces control of the SPP-UX console to your screen and prints the last message stored in the message buffer.

hostname

Identifies the current host's console.

Step 3 Exit the session with the following command:

CTRL-Ec.

The period is part of the command.

Changing a console's connection

Once you have started a console as a watch or a control connection, you can change the connection type with escape characters.

To change a watch window to an active console window, enter:

CTRL-Ecf

To change an active console window to a watch window, enter:

CTRL-cs

Accessing system logs

You can monitor system status via two logs, event_log and consolelog, located in /spp/data on the Exemplar test station.

The event_log file periodically logs system status. Once the file reaches 4MB, the system renames it to event_log.old and creates a new event_log file.

The consolelog file logs data from the SPP-UX System Console window. When the file reaches 2MB, the system renames it to consolelog.old and starts a new consolelog file.

Customizing startup and shutdown processes

To customize the startup/shutdown process you must:

- Create a configuration variable script for enabling, disabling, or modifying the subsystem. Scripts are located in /etc/rc.config.d.
- Create an execution script that is executed at system startup and shutdown. Scripts are located in /sbin/init.d.
- Create associated sequencer link files to control the run order of the execution script. Links are located in the /sbin/rc*N*.d directories where *N* is the number of the associated run level.

The /etc directory contains configuration data and the /sbin directory contains essential system commands.

Do not directly modify execution scripts. Use the process outlined in "Creating a custom script" on page 12 to prevent having your changes overwritten by subsequent SPP-UX releases.

This section:

- Gives an overview of the required elements
- Illustrates starting the HTTP daemon and describes the steps necessary to customize the process

For a thorough review of the new process see the *HP-UX 10.0 File System Layout Whitepaper*, located on your system in /usr/share/doc/filesys.txt.

Execution scripts

An execution script should be named for the product it controls; for instance, if your script controls the HTTP daemon, name it `httpd`.

The execution script must recognize the following arguments:

- `start`
- `stop`
- `start_msg`
- `stop_msg`

Use the template file, /sbin/init.d/template, as a framework to build a new script.

Local scripts should begin with an underscore (_). This prevents your custom script from being overwritten if a future release of SPP-UX contains a product with the same name.

Configuration scripts

Each subsystem uses one configuration script—located in /etc/rc.config.d—that enables or disables the specific subsystem and, in addition, may modify the subsystem. The configuration script's file name duplicates the execution script's file name.

Edit these configuration scripts, rather than execution scripts, with any text editor. In most scripts, the first variable is named the same as the subsystem and indicates whether to enable or disable the subsystem. The following example shows a typical configuration script, which allows you to enable or disable the HTTP daemon.

```
#
# local configuration:
# set product environment to a non-zero value to
# initiate an httpd server at run-level 2.
#
# installed at /etc/rc.config.d/_httpd
#
RUN_HTTPD=1
```

Most scripts provide comment lines—beginning with a pound sign(#)—to describe variables and their values.

Sequencer link files

A run level is an SPP-UX state of operation in which a specific set of processes is permitted to run. The /sbin/rc*N*.d (where *N* is the number of the associated run level) directories contain only symbolic links—called sequencer links—to the appropriate /sbin/init.d execution scripts. Upon proceeding to different run levels, the /sbin/rc script runs the symbolic links that, in turn, run the execution scripts.

Each run level has its own /sbin/rc*N*.d directory. For example, /sbin/rc0.d contains links to execution scripts invoked when your system enters run-level 0. The /sbin/rc0.d directory is used for both 0 and S run levels.

Start and kill links

Since execution scripts start up and shut down their associated subsystems, two links are required in the sequencer directories: a start link and a kill link.The link type indicates whether the link is used to run its execution script in `start` mode or `kill` mode. Table 1 describes link file characteristics.

Table 1 Link file characteristics

Link type	Begins with	Invoked with the...
Start link	`S`	`start` argument at system boot or when moving to a higher run level.
Kill link	`K`	`stop` argument at system shut down or when moving to a lower run level.

For instance, links beginning with `S` located in /sbin/rc1.d are read and executed when your system proceeds to a higher run level, run-level 2 or above. When you transition to a lower run level, run-level 0, links beginning with `K` are read and executed.

Link name syntax

The following file name illustrates sequencer link name syntax:

```
/sbin/rc2.d/S999_httpd
```

where

`rc2.d`

Designates the directory's associated run level, in this case run-level 2.

`S`

Designates the sequence link type. `S` indicates a `start` link. A `K` in this field designates a kill link.

`999`

Designates the sequence number that dictates link execution order. Valid numbers range from 001 through 999. Links are executed in sequence, using the entire link file name, at:

- System start up
- System shut down

`_httpd`

Duplicates the execution script's name. The initial underscore (_) indicates this is a custom script not supplied with SPP-UX.

Link numbering and locations

Subsystem kill scripts do not have the same sequence number as their start script counterparts. Start scripts in /sbin/rc*N*.d have kill script counterparts in /sbin/rc(*N*-1).d. For example, the

licensing start-up link /sbin/rc2.d/S200licensing has a shut down counterpart called /sbin/rc1.d/K800licensing.

Interdependent subsystems

If two subsystems must be started in a given order due to dependencies, the kill script counterparts must be numbered so that the subsystems are stopped in the opposite order. The following example illustrates that foo(S900foo) starts before uses_foo (S950uses_foo). At shut down, uses_foo (K050uses_foo) stops before foo (K100foo).

```
/sbin/rc1.d          /sbin/rc2.d
K050uses_foo         S900foo
K100foo              S950uses_foo
```

Noninterdependent subsystems

A standard rule is that a subsystem's start and kill sequence numbers added together should equal 1000. Use this rule to help select sequence numbers when customizing the start-up and shut down processes.

If specific ordering is not a concern and the subsystems can be started at any point after system boot and initialization, use the run levels and sequence numbers listed in Table 2.

Table 2 Custom script run levels and sequence numbers

Link type	Run level	Directory	Sequence range
Start link	2	/sbin/rc2.d	900 - 999
Kill link	1	/sbin/rc1.d	100 - 199

Creating a custom script

The following instructions create an execution script, configuration script, and sequencer links required to add the HTTP daemon to your system startup and shut down processes:

Step 1 Log in as root.

Step 2 Create the execution script in /sbin/init.d.

```
cp /sbin/init.d/template /sbin/init.d/_httpd
```

The file must be readable and executable by root.

The script begins with an underscore (_) to prevent your custom script from being overwritten if a future release of SPP-UX contains a product with the same name.

Step 3 Edit the template file to reflect the correct path.

The following example reflects changes made to the template file in order to correctly start and stop the HTTP daemon.

```
#!/sbin/sh
#
# httpd startup: Startup and kill script for the httpd
#
PATH=/sbin:/usr/sbin:/usr/bin:/usr/local/bin
export PATH
if [ -r /etc/rc.config.d/_httpd ]; then
        . /etc/rc.config.d/_httpd
fi
case "$1" in
        "start_msg") echo "Starting httpd" ;;
        "start") if [ "$RUN_HTTPD" -ne 0 ] ; then
                /usr/local/http/httpd_start && exit 0
                fi
                exit 2 ;;
        "stop_msg") echo "Stopping httpd" ;;
        "stop") if [ "$RUN_HTTPD" -ne 0 ] ; then
                /usr/local/httpd/httpd-stop && exit 0
                fi;;
esac
exit 0;
```

Step 4 Create a configuration script in /etc/rc.config.d.

The file must be readable and executable by root.

You can use existing scripts in /etc/rc.config.d as a model for your script. The following sample shows the configuration script for starting the HTTP daemon:

```
#
# local configuration:
# set product environment to a non-zero value to
# initiate an httpd server at run-level 2.
#
# installed at /etc/rc.config.d/_httpd
#
RUN_HTTPD=1
```

The configuration script's file name duplicates the execution script's file name. Comment lines begin with a pound sign (#).

Step 5 Create a start link in the appropriate /sbin/rc*.d directory.

The link must be readable and executable by root.

In the following example:

```
ln -s /sbin/init.d/_httpd /sbin/rc2.d/S999_httpd
```

the httpd daemon starts when the system moves to run-level 2 from run-level 1. The sequence number indicates it is the last product started in run-level 2.

Step 6 Create a stop link in the appropriate /sbin/rc*.d directory.

The link must be readable and executable by root.

In the following example:

```
ln -s /sbin/init.d/_httpd /sbin/rc1.d/K001_httpd
```

the httpd daemon stops when the system moves to run-level 1 from run-level 2. The sequence number indicates it is the first product stopped upon transition to run-level 1.

Starting SPP-UX

You must start up, or boot, SPP-UX when the operating system has been completely shut down or after you have partially shut down the operating system to perform system administration tasks. This section describes:

- Starting SPP-UX
- Reviewing the state of the file system
- Restarting SPP-UX

Starting SPP-UX

The process begins when you power up the Exemplar Technical Server. OBP tests the hardware, provides a complete and accurate description of all available hardware, and boots the operating system.

OBP then passes control to the /sbin/init process, which sequentially executes the contents of /etc/inittab. The inittab file executes the /sbin/bcheckrc and /sbin/rc scripts to check your file system and initialize the system.

The operating system continues to boot and displays additional information about your system. After the boot process completes, an SPP-UX login prompt appears in the System Console window.

Reviewing the state of the file system

During the start-up process, the /sbin/bcheckrc script executes `/usr/sbin/fsclean`. This command determines the shut down status of the system and returns three possibilities:

- Proper file system shut down

  The startup process continues and you see the following message:

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

- Improper file system shut down

  The start-up process is interrupted and you see:

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

  At this point, the system runs `/usr/sbin/fsck` in a mode that corrects certain inconsistencies in the file systems without your intervention and without removing data. The `fsck` command does one of the following:

  - Repairs and reboots the system, incorporating the changes

  - Prompts you to run the `fsck` command manually. If you need to run `fsck` manually, see the fsck(1m) manpage
- Other errors detected

  An error message displays (for example, unable to open a specified device file), the start-up process ends, and you need to solve the problem.

Restarting SPP-UX

To restart the system after a reboot or hang, you must first determine if the system is in single-user state (run-level s.) If it is not in single-user state, you must bring the system to single-user state before you can restart your system safely. Characteristics include:

- The only access to the system is through the console by the root user
- The only processes running on the system are the
  - Shell on the console
  - Background daemon processes started by `/sbin/rc`
  - Processes that the root user invokes

If SPP-UX is in single-user state

Complete the following steps to restart your system:

Step 1 Log in to the test station as root.

Step 2 Reboot the system with the `reboot` command. Enter:

```
reboot
```

If SPP-UX is not single-user state

Complete the following steps to restart your system:

Step 1 Log in to the test station as root.

Step 2 Change to the root directory. Enter:

```
cd /
```

Step 3 Bring the system to single-user state with the `shutdown` command. Enter:

```
shutdown
```

Step 4 Reboot the system with the `reboot` command. Enter:

```
reboot
```

Setting run levels

A run level is an SPP-UX state of operation in which a specific set of processes is permitted to run. These processes and default run levels are defined in the /etc/inittab file. Tasks covered in this section include:

- Creating a new run level
- Changing system run levels
- Entering the system administration run level
- Exiting the system administration run level

Run level stages have predetermined behaviors. Table 3 describes each run level.

Table 3 SPP-UX run level descriptions

Run level	Description
s	Single-user mode
0	Halted run level
1	Boot-time system configuration
2	Normal operating mode or multiuser mode
3	Exports NFS file systems
4/5/6	Not currently used

The /etc/inittab file defines default run levels and processes specific to your system. For more information see the inittab(4) and init(1M) man pages.

The following /etc/inittab example calls for a console and the `initdefault` run level is run-level 3:

```
init:3:initdefault:
stty::sysinit:/sbin/stty 9600 clocal icanon echo opost onlcr ienqak ixon \
icrnl ignpar < /dev/systty
brc1::bootwait:/sbin/bcheckrc </dev/console >/dev/console 2>&1  # fsck, etc.
sqnc::wait:/sbin/rc </dev/console >/dev/console 2>&1            # sys init
cons:0123456:respawn:/usr/sbin/getty -h console console         # sys console
```

Creating a new run level

Define how you want your system to operate when in a particular run level by creating or changing entries in the /etc/inittab file.

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

Step 1 Log in as root.

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

```
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 to change the `initdefault` entry in your test version to `s`.

By changing the initdefault entry to `s`, your system boots up in run-level `s`.

Step 4 Change to run-level 2 after booting. Enter:

```
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 a test version, replace the `s` with `2` after you complete testing. Run-level `s` is for system maintenance only.

Step 5 Test the changes you have made.

Step 6 Restore the original `initdefault` value after thoroughly testing your changes.

If you do not have a working state for the `initdefault` state, you may not be able to boot your system.

Changing system run levels

This section contains general procedures for changing your system from one run level to another.

Changing run levels while users are logged on kills their processes if the run level you are moving to does not contain entries in /etc/inittab for their terminal. Warn users currently logged in with the `write` or `wall` commands.

Step 1 Log in to the console as root.

Step 2 Warn all users who are currently logged in before you change run levels.

Use the `write` or `wall` commands to communicate with your users. The `wall` command immediately sends your message to the terminal of each user on the system.

You do not need to ask your users to log off if:

- Each `getty` terminal entry has the new run level in its rstate field
- The rstate field is empty, which implies all numbered run-level `s`

Their processes will not be killed unless your new run level is run-level `s`.

Step 3 Change run levels. Enter:

`/sbin/init` *new_run_level*

where *new_run_level* is the number of the run level you want to enter.

Entering the system administration run level

Many system maintenance tasks require the system to be in single-user mode, run-level `s`, to ensure no one else is on the system while you are performing those tasks. In this run level:

- The only access to the system is through the console by the root user
- The only processes running on the system are the:
  - Shell on the console
  - Background daemon processes started by `/sbin/rc`
  - Processes that you invoke

Commands requiring an inactive system—such as `fsck`—should be invoked in run-level `s`.

Use `shutdown` to change your system's run level from any numbered run level to run-level `s`. `shutdown` kills all nonessential processes and brings the system safely to run-level `s`, without leaving system resources in an unusable state.

See the shutdown(1M) man page for a complete description of the shutdown process and available options.

Step 1 Log in to the console as root.

Step 2 Begin the shutdown process. Enter:

`shutdown`

See "Shutdown considerations" on page 21 for more information on shutdown procedures.

Exiting the system administration run level

To change your system's run level from run-level `s`, reboot your system with the `reboot` command. Enter:

```
/usr/sbin/reboot
```

The system automatically reboots to multiuser mode.

Stopping SPP-UX

Typically, you shut down the system to:

- Put it in single-user state so you can:
  - Update the system
  - Check file systems
- Turn it off in order to perform a task such as installing a new disk drive

Caution

Never stop the system by turning off the power. Stopping the system improperly can corrupt your file systems. Use `shutdown`.

Power outage considerations

If you know power is going out soon, shut down the computer and turn off the power.

Local power failures

A local power failure halts your computer by affecting its central bus. If a local power failure occurs, turn off all computer equipment affected by the power failure until power is completely restored. An electrical surge, such as when power comes back on, could seriously damage hardware that has been left turned on.

Remote power failures

The following power failures do not affect the system as a whole, unless the remote devices provide a vital system resource:

- A remote power failure—Affects a remote bus
- A device power failure—Affects a device

Shutdown considerations

Only the system administrator or a designated superuser can shut down the system.

The `/sbin/shutdown` command:

- Warns all users using a grace period you specify to log off the system
- Halts daemons
- Kills unauthorized processes
- Unmounts file systems
- Puts the system in single-user mode
- Writes the contents of the I/O buffers to a disk

Do not run `shutdown` from a remote system via `rlogin` if you use a network service. The shutdown process logs you out prematurely and returns control to the console. Run `shutdown` from the System Console window on the test station.

See the shutdown(1M) man page for a complete description of the shutdown process and available options.

Rebooting or shutting down the system

To reboot or shut down your Exemplar Technical Server, perform the following steps:

Step 1 Log in to the console as root.

Step 2 Change to the root directory. Enter:

```
cd /
```

Step 3 Shut down the system `shutdown`. Enter:

```
shutdown
```

The following example shows output from the `shutdown` command:

```
Do you want to send your own message?(You must respond with 'y' or
'n'.): y

Type your message.
End with a new-line and EOF character ([CTRL]-[D]).

system shutting down in 1 minute


Broadcast Message from root(ttyp7) Mon Oct  7 20:35:56...
system shutting down in 1 minute

All processes will be terminated in 60 seconds.


Waiting a grace period of 60 seconds for users to logout
Do not turn off the power or press reset during this time.

Broadcast Message from root(console) Mon Oct  7 21:15:19...
SYSTEM BEING BROUGHT DOWN NOW ! ! !


Do you want to continue?(You must respond with 'y' or 'n'.): y
```

Progress messages detailing system shutdown activities print to your terminal. Upon reaching run-level 0, the system:

- Restarts in single-user mode

- Displays a root prompt

Step 4 Bring the system to a complete stop with the `reboot` command. Enter:

```
reboot -h
```

You can now turn off the power or restart the system.

Controlling system access 2

Securing your data against deliberate, unauthorized access is one reason for controlling access to your system. Three levels of access control to your system are:

- Controlling user accounts
- Controlling groups
- Controlling file access

Terms

Terms used in this chapter include:

GID (group_ID)

A unique number associated with each group, that identifies the group to SPP-UX. These GID numbers are defined in the /etc/group file.

UID (user_ID)

A unique number that SPP-UX uses to identify a particular user. The UID number zero ("0") identifies the superuser. UID numbers between 1 and 99 are reserved for SPP-UX subsystems. UID numbers above 99 are used for regular users.

effective group

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

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 has access to the file. Users can be members of more than one group.

group ownership

The secondary ownership associated with each file, associating the file with a group.

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 creates the file.

primary 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 in to the system, they are affiliated with their primary group.

user account

The environment created on the system to allow 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.

Changes specific to this release

The following programs are located in /usr/sbin:

- `useradd`
- `userdel`
- `usermod`
- `groupadd`
- `groupdel`
- `grpck`

You cannot use `sam`, Hewlett-Packard's System Administration Manager utility, to setup or maintain user accounts or groups.

Users

Each user is defined by an entry in the file /etc/passwd. Use the /usr/bin/vipw command to modify the /etc/passwd file. The vipw command guarantees exclusive access to /etc/passwd. The vipw, chsh, chfn, and passwd commands create an /etc/ptmp file when granting access to the /etc/passwd file. Set your EDITOR environment variable to vi to use vipw.

Caution

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 removing their entry from the /etc/passwd file.

The following sections show you how to add, remove, and change a typical user account and groups. The user's login name is pat. Primary and secondary groups are user and test.

Checking or changing useradd defaults

Before you begin, check the defaults assigned to your system's copy of useradd. If you do not want to accept the preset defaults you will need to change the settings.

Checking defaults

To check the defaults assigned to useradd on your system, use the -D option.

Step 1 Log in as root.

Step 2 Check the defaults with useradd. Enter:

```
useradd -D
```

```
GROUPID  20
BASEDIR  /home
SKEL     /etc/skel
SHELL    /sbin/sh
INACTIVE -1
EXPIRE
```

This sample output indicates the default GID, home directory, skeleton directory, command shell, number of days permitted inactive, and expire date.

Changing defaults

To change the defaults assigned to `useradd` on your system, use the `-D` option combined with the appropriate option. Valid options include:

`-g` *GID*

Changes default group ID. Can be an integer or character string.

`-d` *home_dir*

Changes default home directory location.

`-k` *skeleton_dir*

Changes skeleton directory. The default skeleton directory—/etc/skel— contains the following login files: .cshrc, .exrc, .login, and .profile.

`-s` *shell_program*

Changes the login shell. Default is /bin/sh.

`-f` *#_of_days*

Changes the default maximum number of days of continuous inactivity allowed before the login is invalid. Normal values are positive integers, -1 deactivates this option.

`-e` *expire*

Specifies the date on which this login can no longer be used. Once expired, this login is inaccessible. This option is used to create temporary logins. *expire* may be any format, except a Julian date. For example, a date may be entered in either of the following formats:

```
July 13, 1993
7/13/93
```

A value of four apostrophes ("") disables the expire date status.

Step 1 Log in as root.

Step 2 Change the defaults with `useradd -D`. Enter:

```
useradd -D -g 50
```

This sample command changes the default group ID to group 50. Any accounts subsequently created without a specifically assigned group ID belong to group 50.

Adding a user

To add a new user, pat, with `useradd`:

Step 1 Log in as root.

Step 2 Determine the information you want to assign to the new user.

Table 4 outlines `useradd`'s associated options. See the useradd(1M) man page for more information.

Table 4 Associated options for `useradd`

Information	Description	Option
UID	User's numeric system identification number (UID)	`-u`
Primary GID	User's primary group membership (GID); can be an integer or character string	`-g`
Nonunique UID	(Optional) Allows the UID to duplicate other system UIDs	`-o`
Secondary GIDs	(Optional) User's secondary group memberships, comma separated; can be an integer or character string	`-G`
Login shell	User's default command interpreter	`-s`
Login name	(Required) User's unique login name	N/A
Home directory	Location of users home directory	`-d`
Create home directory	(Optional) Makes the user's home directory. If you do not supply the `-m` option you must create the home directory with SPP-UX commands	`-m`
Comments	(Optional) Description of login. Can be any text string	`-c`
Login files	(Optional) Provides minimum files required to log in. The default (/etc/skel) includes .cshrc, .login, .exrc, and .profile	`-k`
Continuous inactivity allowed	(Optional) Designates the maximum number of days account can be inactive—no logins—before becoming invalid. Can be any positive integer, `-1` disables this feature	`-f`
Expire date	(Optional) Sets the expiration date for temporary accounts. Can be any format except Julian date. Four apostrophes ("") disable this feature	`-e`

Step 3 Add the user with `useradd`.

For instance, entering:

```
useradd -u 1116 -g users -G test -d /home/pat -s /bin/csh -m pat
```

produces the following system information:

`-u 1116`

Designates the UID as 1116.

`-g users`

Designates the primary group as users.

`-G test`

Designates the secondary group as test.

`-d /home/pat`

Assigns the home directory as /home/pat.

`-s /bin/csh`

Assigns the login shell as /bin/csh.

`-m`

Makes the directory and copies default startup files to the directory specified by the `-d` option.

Step 4 Set the password for the new account with `passwd`.

The `useradd` utility places an asterisk (*) in the password field of the new /etc/passwd entry. This prevents logins from the local machine. Use the passwd(1) command to set a password for the account. In this example enter:

```
passwd pat
```

The passwd(1) command prompts you for the new password twice. Notify the new user of the password you assigned to the account and request they change the password as soon as possible.

Removing a user

To remove a user using `userdel`:

Step 1 Log in as root.

Step 2 Delete the account with `userdel`.

```
userdel -r pat
```

This example removes the user pat from login related files. The `optional -r` argument removes pat's home directory from the system.

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

Changing user information

To change a user's account information, use the `usermod` command. This example changes the user pat's primary group id (GID) to the test group.

Step 1 Log in as root.

Step 2 Determine the account information you want to change.

Table 5 lists information you can change and associated options. See the usermod(1M) man page for more information.

Table 5 Information you can change with `usermod`

Information	Description	Option
UID	User's numeric system identification number (UID)	`-u`
Nonunique UID	Allows the UID to duplicate other system UIDs	`-o`
Primary GID	User's primary group membership (GID); can be an integer or character string	`-g`
Secondary GIDs	User's secondary group memberships, comma separated; can be an integer or character string	`-G`
Home directory	Location of user's home directory	`-d`
Move home directory	Moves the user's home directory to the directory specified with the `-d` option	`-m`
Login shell	Users default command interpreter	`-s`
Comments	Description of login. Can be any text string	`-c`
New login name	User's unique login name	`-l`
Continuous inactivity allowed	Designates the maximum number of days account can be inactive—no logins—before becoming invalid. Can be any positive integer, `-1` disables this feature	`-f`
Expire date	Sets the expiration date for temporary accounts. Can be any format except Julian date. Four apostrophes ("") disable this feature	`-e`

Step 3 Change the information with `usermod`. Enter:

```
usermod -g test pat
```

Groups

Users on your system can be divided into various working groups. Members of a group may share, and yet protect, files from access by users who are not members of that group.

A user can be a member of more than one group. A group can have a maximum of 200 members.

Group information is defined in /etc/group and /etc/logingroup files. Each entry in the /etc/group file defines the:

- Group name
- Numerical group identifier or *group_ID*—Uniquely identifies entries in /etc/group and /etc/logingroup
- Group members—A comma-separated list of user login names
- Encrypted password (optional)

Tasks covered in this section cover:

- Adding a group
- Removing a group
- Changing a user's primary group
- Adding a user to a group
- Removing a user from a group

Changing group membership temporarily

Users can temporarily change their current group affiliation by using `newgrp`. The new group is referred to as the effective group for that 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 primary group membership by specifying no options to the `newgrp` command.

Information you need to add a group

You need the following information to add a group:

- Group name
- GID
- (Optional) Users to include in the new group

Adding a group

To add a group, use `groupadd`. This example adds the group operator with GID of 201. See the groupadd(1M) man page for more information.

Step 1 Log in as root.

Step 2 Add the group with `groupadd`. Enter:

```
groupadd -g 201 operator
```

Removing a group

To remove a group, use `groupdel`. This example removes the group operator. See the groupdel(1M) man page for more information.

Step 1 Log in as root.

Step 2 Remove the group with `groupdel`. Enter:

```
groupdel operator
```

Changing a user's primary group

A user can be a member of multiple groups, but only one of those groups is the user's primary group.

In addition to being listed as a member of the group in the file /etc/group, an entry exists in /etc/passwd indicating the user's primary group. When users first log in to the system, they are affiliated with their primary group.

To change a user's primary group use the `-g` option of the `usermod` command. This example changes the user pat's primary group to the test group. See the usermod(1M) man page for more information.

Step 1 Log in as root.

Step 2 Change the information with `usermod`. Enter:

```
usermod -g test pat
```

Adding a user to a group

To add a user to a group use the `-G` option of the `usermod` command. This example adds the user pat to the operator group. See the usermod(1M) man page for more information.

Step 1 Log in as root.

Step 2 Add the user with `usermod`. Enter:

```
usermod -G operator pat
```

Removing a user from a group

You cannot remove a user from their primary group. Change the user's primary group before attempting to remove them.

To modify a group's membership you must use a text editor and `grpck`.

Step 1 Log in as root.

Step 2 Copy the /etc/group file to /etc/group.old so that it is easy to undo any mistakes that you might make:

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

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

Step 3 Edit the /etc/group file to remove the user from the appropriate group entry.

Save and exit the file when you are finished making changes.

Step 4 Check the group file with `grpck`. Enter:

```
/usr/sbin/grpck
```

The `grpck` command checks the /etc/group file for inconsistencies and verifies all entries. This verification includes a check of the number of fields, group name, group ID, and whether all login names appear in the password file.

Accessing files

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

File and ownership access permissions

There are three types 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, execute permissions determine who can run the program. For directories, execute access allows listing the directory contents.

There are three types of file access:

Owner

Usually the person who created the file (unless ownership has since been changed using the `chown` command.)

Group

Members of the group to which the file belongs.

Other

All other users on your system.

Changing access privileges

Three commands change file access privileges:

`chmod`

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`

Changes file ownership. In order to change the owner, you must own the file or have superuser privileges.

`chgrp`

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.

The system assigns default file permissions—governed by your `umask`—whenever you create a new file or directory. Unless set up otherwise, your default `umask` setting is 0, which means that new files you create have read/write permission for everyone (666 or -rw-rw-rw-) and new directories you create have read/write/search permission for everyone (777 or drwxrwxrwx).

For more information see the ll(1), setprivgrp(1M), and umask(1) man pages.

Managing disks

3

Once you physically install a disk, you must reboot your Exemplar S-Class and X-Class Technical Servers in order for OBP to create a device path. After booting SPP-UX, you must map the disk to a specific logical device name using `diskutil`. You also use `diskutil` to configure disk stripes and partitions.

After configuring the disk you need to monitor and perhaps impose quotas on system users.

This chapter details:

- Disk basics
  - Naming conventions
  - Device files
- Using `diskutil` to configure disks
- Monitoring disk usage
- Using disk quotas

Changes specific to this release

The `diskutil` program is located in /usr/sbin.

Disk basics

After you install a disk and boot SPP-UX, you must map the disk device to a logical device name. This process:

- Creates device files
- Writes the logical device name to a physical device

Device files

Physical devices are accessed by users through device files. A device file is a special file that represents an I/O device. Physical devices are accessed through these special files in the same way an ordinary file is accessed. There are two types of device files:

- Character device files—Access devices that typically transmit only one character or line at a time, for instance, terminals and printers. These files are located in the /dev/rdsk directory. Character device files are also called raw device files
- Block device files—Access buffered devices that transfer information in blocks, like disk and tape devices. These files are located in the /dev/dsk directory

You need both types of device files for disk devices and stripe devices.

Disk devices

The `diskutil` command creates the required files when you map the disk device. Disk device file names begin with `sd`.

Stripe devices

The `diskutil` command creates the required files when you create the stripe. Stripe device file names begin with `stripe`.

Is your disk mapped?

When your Exemplar Technical Server boots SPP-UX, status messages print to the System Console window. The status message for a mapped drive looks like:

```
[62000001 001e2500] scsi disk: disk 0:0:4:0 attached mapped to sd17
```

The status message for an unmapped drive looks like:

```
[62000001 001e3288] scsi disk: disk 0:0:5:0 not mapped
```

Also, the /spp/data/event_log file keeps a record of all drives mapped during the boot process.

Determining drive mappings

You can determine your Exemplar Technical Server's disk drive mappings:

- During the SPP-UX boot process, by checking status messages printed to the System Console.

  The status message for a mapped drive looks like:

```
[62000001 001e2500] scsi disk: disk 0:0:4:0 attached mapped to sd17
```

  The status message for an unmapped drive looks like:

```
[62000001 001e3288] scsi disk: disk 0:0:5:0 not mapped
```

- After the boot process, by checking your system's /spp/data/event_log file for a record of all drives mapped during the boot process. These entries duplicate the status messages printed to your screen during the SPP-UX boot process.
- By using `diskutil`'s `show disk` option. For example:

```
DiskUtil: show disk
SD 0:0:2:0 mapped to sd0
SD 0:0:3:0 mapped to sd9
```

Mapped drive naming convention

Once the disk device is mapped, the physical device names correspond to logical device names. The device keeps the same logical device name no matter which physical hardware ID is subsequently assigned.

The physical device name looks like:

```
0:0:4:0
```

Table 6 explains the mapping convention.

Table 6 Physical device name map

Field	Maps to...
First	Node number
Second	Controller slot
Third	SCSI ID
Fourth	Logical unit number

A typical logical device name looks like:

`sd2a`

where

`sd`
: Designates a SCSI disk.

`2`
: Specifies the device number. Starts at zero (0) and increments by one for every disk on the system. Device numbers do *not* have to be sequential.

`a`
: Specifies the partition. Valid characters include `a` through `p`.

Using diskutil

The `diskutil` command helps you:

- Install and use disks
- Create stripes and partitions

You can access `diskutil` via the command line or as an interactive shell. The examples contained in this chapter use `diskutil` interactively.

To view the general help screen enter:

```
DiskUtil: help

        The available commands are:
          Create    Destroy  Exit      Force     MAKe      MAP       Quit
          SELect    SET      SHow      UNMap     UNSet

        To obtain more information about a command, type:

         Help <command-name>

       Each command keyword is always shown as a sequence of upper
```

and lowercase letters. The uppercase letters at the beginning of each keyword show the shortest abbreviation acceptable for the command keyword.

To exit or terminate a diskutil session, enter Exit or Quit at the diskutil prompt.

Setting a disk's logical unit name

To set the logical unit name for an unmapped disk device, use diskutil's map disk option.

Step 1 Log in as root.

Step 2 Start diskutil. Enter:

```
/usr/sbin/diskutil
```

Step 3 Determine the physical device's name. Enter:

```
DiskUtil: show disks

SD 0:0:2:0 mapped to sd7
SD 0:0:3:0 mapped to sd0
SD 0:0:4:0 unmapped
SD 0:0:5:0 mapped to sd16
SD 0:0:9:0 mapped to sd17
SD 0:0:10:0 mapped to sd1
```

In this example, the unmapped disk's physical device name is SD 0:0:4:0

Step 4 Map the disk with map disk.

For example, the command:

```
DiskUtil: map disk SD 0:0:4:0 to sd19
```

maps the physical device, SD 0:0:4:0, to logical unit name sd19.

Unmapping a disk

You will not be able to unmap a disk if its partitions are active. You can check to see if a partition is active with the show partitions option, active partitions are marked with an asterisk (*) in the flags column. Unmounting a partition makes it inactive.

To unmap a physical disk use diskutil's unmap disk option.

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Determine the physical device's name. Enter:

```
DiskUtil: show disks

SD 0:0:2:0 mapped to sd7
SD 0:0:3:0 mapped to sd0
SD 0:0:4:0 mapped to sd19
SD 0:0:5:0 mapped to sd16
SD 0:0:9:0 mapped to sd17
SD 0:0:10:0 mapped to sd1
```

In this example, you want to unmap the `SD 0:0:4:0` disk.

Step 4 Unmap the disk with `unmap disk`. Enter:

```
DiskUtil: unmap disk SD 0:0:4:0
```

Disk partitions

Once your disk is mapped you can separate it into partitions. Partitions have no size limit; a single disk may only have 15 partitions.

The `diskutil` commands allows you to set the following partition attributes:

- Size—Expressed in bytes, must be a multiple of 1024 bytes
- Position—Expressed as
  - `offset` *value*—Expressed in bytes, must be a multiple of 1024 bytes
  - `after` *partition*
  - `before` *partition*
- Description—Purpose for the partition
- Flag—Set after the partition is created. Required if the partition is to be used as:
  - Crashdump
  - Default pager
  - Raw

Subsequent sections detail how to:

- Display partitions
- Create partitions
- Set disk partition flags and descriptions
- Delete partitions

Displaying disk partitions

To display a disk's partitions use the `show partitions` option.

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Show the available disks. Enter:

```
DiskUtil: show disks

SD 0:0:2:0 mapped to sd7
SD 0:0:3:0 mapped to sd0
SD 0:0:4:0 mapped to sd19
SD 0:0:5:0 mapped to sd16
SD 0:0:9:0 mapped to sd17
SD 0:0:10:0 mapped to sd1
```

Step 4 Select the disk you want to view with `select disk`.

```
DiskUtil: select disk sd19
```

In this example, you want to see the disk labeled `sd19`.

Step 5 Show the partitions. Enter:

```
DiskUtil: show partitions

Logical disk name: sd19
partition table: (space available for file systems = 2098759)
part    offset      size    | partition description          | flags
-------------------------------------------------------------------
 a:          0K   819200K  |Root and /usr filesystem        | *
 b:     819200K  1048576K  |Default Pager Partition         | *D
 c:    1048576K    40960K  |Crashdump Partition             | C
```

The disk in this example, `sd19`, had three partitions, two are active (marked with a * in the flags field.)

Creating a disk partition

A disk drive may have only 15 partitions. This example creates partition a with a size of 102400 K starting at the beginning of the disk.

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Show available disks. Enter:

```
DiskUtil: show disks
```

```
SD 0:0:2:0 mapped to sd7
SD 0:0:3:0 mapped to sd0
SD 0:0:4:0 mapped to sd19
SD 0:0:5:0 mapped to sd16
SD 0:0:9:0 mapped to sd17
SD 0:0:10:0 mapped to sd1
```

Step 4 Select the disk.

```
DiskUtil: select disk sd19
```

In this example, you selected the disk labeled `sd19`.

Step 5 Make the partition with `make partition`. Enter:

```
DiskUtil: make partition a size 1000m offset0
```

Setting disk partition flags and descriptions

Use `diskutil`'s `set partition` option to designate a partition's flags and descriptions.

Setting descriptions

The description is human readable and describes the purpose for the partition. You can set it when you create the partition or after creating it. The following example sets the description field for the a partition to `root and /usr files`:

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Show available disks. Enter:

```
DiskUtil: show disks
```

```
SD 0:0:2:0 mapped to sd7
SD 0:0:3:0 mapped to sd0
SD 0:0:4:0 mapped to sd19
SD 0:0:5:0 mapped to sd16
SD 0:0:9:0 mapped to sd17
SD 0:0:10:0 mapped to sd1
```

Step 4 Select the disk.

```
DiskUtil: select disk sd19
```

In this example, you selected the disk labeled `sd19`.

Step 5 Set the description with `set partition`. Enter:

```
DiskUtil: set partition a description "root and /usr files"
```

Setting flags

A flag is required if you use a partition as crashdump, default pager, or raw. Set the flag after your create the partition. The flags field notes the type of partition. There are tree types of flags:

1. Flags set by the system
2. Flags you set that are used by the system
1. Flags you set for information only—not used by the system

Table 7 describes flag options.

Table 7 Partition flag descriptions

Flag	Description	Set by the...
None	Standard filesystem	System
*	Busy filesystem. Can be mounted or in use by the default pager	System
D	Default pager partition. Swap space	Administrator; used by the system
S	Stripe. Partition is part of a stripe	System
C	Crashdump partition	Administrator; used by the system
R	Raw partition	Administrator; information only

This example sets partition c's flag to `crashdump`.

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Show available disks. Enter:

```
DiskUtil: show disks

SD 0:0:2:0 mapped to sd7
SD 0:0:3:0 mapped to sd0
SD 0:0:4:0 mapped to sd19
SD 0:0:5:0 mapped to sd16
SD 0:0:9:0 mapped to sd17
SD 0:0:10:0 mapped to sd1
```

Step 4 Select the disk.

```
DiskUtil: select disk sd19
```

In this example, you want to see the disk labeled `sd19`.

Step 5 Set the flag with `set partition`. Enter:

```
DiskUtil: set partition c flag Crashdump
```

Deleting a disk partition

To delete a partition, remake it with a size of zero.

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Show available disks. Enter:

```
DiskUtil: show disks
```

```
SD 0:0:2:0 mapped to sd7
SD 0:0:3:0 mapped to sd0
SD 0:0:4:0 mapped to sd19
SD 0:0:5:0 mapped to sd16
SD 0:0:9:0 mapped to sd17
SD 0:0:10:0 mapped to sd1
```

Step 4 Select the disk.

```
DiskUtil: select disk sd19
```

In this example, you selected the disk labeled `sd19`.

Step 5 Make the partition with `make partition`. Enter:

```
DiskUtil: make partition a size 0
```

This sets the size of partition `a` to 0.

Disk stripes

A file system's size is limited by the size of the physical disk. Disk striping allows you to:

- Increase size, by combining multiple physical disk partitions into a single logical partition
- Increase throughput, because data writes simultaneously to each partition in the stripe

Although not required, each physical disk partition in a stripe is usually the same size.

Subsequent sections of this chapter detail how to:

- Create a stripe
- Display a stripe
- Select a stripe
- Display stripe information

Creating a stripe

By default, each stripe is created with 8k blocksize. However, the system's buffer cache is 64k. To improve stripe performance, set your stripe's blocksize to 64k.

This section creates a stripe—stripe21—from partitions sd1h, sd33h, and sd5h.

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Create the partitions you want to include in the stripe. See "Creating a disk partition" on page 43 for instructions.

Step 4 Create the stripe. Enter:

```
DiskUtil: create stripe stripe21 Blocksize 64k (sd1h sd33h sd5h)
```

where

`stripe21`

Is the stripe name. The stripe name format is `stripe`*n* where *n* is a decimal integer 0 through 256.

`Blocksize`

Is the blocksize of the stripe. Default is 8k. To improve performance set the blocksize to 64k (the same as the buffer cache.)

`sd1h sd33h sd5h`

Designate the partitions you want to include in the stripe. Your disk partitions may have different names.

Displaying stripes

To display any stripes currently located on your system:

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Show the stripes. Enter:

```
DiskUtil: show stripe
```

```
stripe Num    Valid/invalid      nodes   partitions
stripe3          valid           0       2f    3f
stripe10         valid           0       3g    0g    2g
stripe11         valid           0       3h    0h    2h
stripe12         valid           0       3i    0i    2i
```

```
stripe13        valid           0       3j      0j      2j
stripe20        valid           1       1g      33g     5g
stripe21        valid           1       1h      33h     5h
stripe22        valid           1       1i      33i     5i
stripe23        valid           1       1j      33j     5j
```

Selecting a stripe

You must select a stripe to perform administration tasks or to display stripe information.

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Select the stripe. Enter:

```
DiskUtil: select stripe stripe21
```

This example selects stripe21.

Displaying stripe information

This section selects a stripe—stripe21— and displays information about its size, blocksize, and partitions.

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Select the stripe with `select stripe`. Enter:

```
DiskUtil: select stripe stripe21
```

Step 4 Display the stripe's information with `show stdisk`. Enter:

```
DiskUtil: show stdisk

stripe21: unit = 21, node = 1, flags = 0x1, serial num = 56581
    size = 1310688, stripe blocksize = 16384, disk blocksize = 1024
    partitions are:
        1h 33h 5h
```

Using disk quotas

The disk quota system is an optional feature that provides a mechanism to control use of disk space by users. You can set quotas for any or all users on any or all file systems.

You can restrict the amount of disk space available to a user, the number of files (inodes) a user can own, or both. If both limits are set, the user is restricted by whichever limit is exceeded first.

You can set soft and hard limits for both the amount of disk space and the number of files.

Disk quota use and limits are stored in a file named quotas located on the mount point of the file system where quotas are imposed. The data in the quotas file is an array of structures, indexed by UID, with one structure for each user on the system regardless of whether the user has a quota on the file system. Do not change this file name because several user-level utilities depend on it. Also, do not copy the quotas file.

Setting up quotas

For this example, quotas are set on the /home directory for users pat, bob, and janet. In order to set up disk quotas complete the following steps:

Step 1 Decide which file systems require quotas.

Typically, set disk quotas on file systems that otherwise become full without limitations. For example, to prevent users from using /tmp or /var/tmp as storage, set the soft limits small and the time limits short.

Step 2 Decide on the block or inode limits for each user in the file system where you will implement quotas.

You can impose any combination of hard and soft limits for each user, and limits can be different for each user, but, typically, set block or inode limits, not both.

A limit set to zero is disabled. Disabling all limits for a user disables the entire quota for that user; that is, no quotas are imposed.

Step 3 Decide on the time limits you want to apply to the file systems where you intend to implement quotas.

The time limit specifies the length of time users have to reduce the numbers of blocks or files once they have reached their soft quota limits.

A single time limit applies to all file systems.

The default time limit for both file system blocks and files is seven days.

Step 4 Log in as root.

Step 5 Check to make sure that the file systems to receive quotas are mounted. Enter:

```
bdf
```

If the filesystem appears in the output, it is currently mounted.

If the file system is not mounted, mount it using the `mount` command. For example, to mount the /home file system enter:

```
mount /home
```

Step 6 Create an empty file called quotas in the root of each file system for which you want to enforce quotas with `cpset`.

In this instance, the root directory is /home, so you enter:

```
/usr/bin/cpset /dev/null /home/quotas 600 root nbin
```

For details, see the cpset(1M) man page.

Step 7 Setup a prototype quota file with `edquota`. Enter:

```
/usr/sbin/edquota masterfile
```

In this example, `masterfile` contains the quota information you assign to all users. Your EDITOR environment variable determines the editor invoked. The default is `vi`.

- Enter the quota limits you wish to enforce, one line for each file system with a quotas file. For instance:

  ```
  fs /home blocks (soft = 10000, hard = 12000)
  ```

- Update the quotas file by saving your changes and exiting the editor

Step 8 Apply the prototype user's limits to other users of the /home file system with the edquota -p command. Enter:

```
edquota -p masterfile pat bob janet
```

In this example, the quota limits set in the prototype file `masterfile` are assigned to the users pat, bob, and janet.

Step 9 Set soft time limits with `edquota -t`. Enter the time limits you wish to enforce, one line for each file system with a quotas file.

```
edquota -t
```

The following example accepts the 7 day default by entering zeros in the time limit fields.

```
fs /home blocks time limit = 0, files time limit = 0
```

Update the quotas file by saving your changes and exiting the editor.

Starting quotas

You can start the quotas you have set up:

- At boot time—By adding the quota option to the file system entry in the /etc/fstab file. This is the preferred method. For example:

  ```
  /dev/dsk/sd2b   /home   hfs rw,suid,quota 0 1
  ```

- While the system is up—If you cannot unmount and remount the file system, because it is in use, you can use the `quotaon` command

For this example, quotas are set on the /home directory for users pat, bob, and janet.

Step 1 Log in as root.

Step 2 If not already mounted, mount the file system for which you will impose disk quotas.

```
mount /dev/dsk/sd2b
```

Step 3 Start quotas with the quotaon(1M) command. Enter:

```
/usr/sbin/quotaon -v /home
```

The `-v` option generates messages listing each file system affected.

Turning off quotas

You can turn off quotas by:

- Unmounting the quota enabled file system with the umount(1M) command. This is the preferred method
- Using the quotaoff(1M) command. However, using quotaoff causes the system to discontinue tracking quota information and causes existing quota information to become incorrect

Checking limits and usage

Use the following commands to help you check disk usage and quotas:

- `/usr/sbin/repquota`
- `/usr/sbin/quot`
- `/usr/bin/quota`

The `repquota` command checks disk usage and quotas for a filesystem. To show the usage for each user of the file system /home:

```
/usr/sbin/repquota /home
```

```
/dev/dsk/sd2b (/home):
                        Block limits                        File limits
User            used    soft    hard    timeleft     used    soft    hard    timeleft
pat     --      59      100     200                  24      30      40
janet   +-      199     100     200     1.7 weeks    10      30      40
bob     --      63      100     200     1.4 weeks    9       30      40      1.4 weeks
```

To display usage for each user on all file systems with quotas enabled listed in the /etc/fstab file use:

```
/usr/sbin/repquota -a
```

Use `/usr/sbin/quot` to check the ownership of 1024-byte blocks for a file system.

```
/usr/sbin/quot /home
```

```
/dev/dsk/sd2b (/home):
 2843   janet
 2429   bob
  164   pat
```

Use `/usr/bin/quota` to generate a report only for users who have exceeded their quotas. The `-v` option displays all user's quota information.

Soft limit and hard limit guidelines

Once a user exceeds the soft limit, the system issues the following warning message to the user's terminal:

```
WARNING: disk quota (/home) exceeded
```

but allows the user to continue working for a limited time.

Once the time limit or hard limit is reached, the user is not allowed to write any more data to the disk. That is, requests for space or attempts to create a file fail. Hard limits and time limits cannot be exceeded.

On the first failure, the system issues the following message to the user's terminal upon reaching a block limit:

```
DISK LIMIT REACHED - WRITE FAILED
```

The following message is written to the user's terminal upon reaching a file limit:

```
FILE LIMIT REACHED - CREATE FAILED
```

Only one message is sent each time a hard limit is reached, no matter how many failures occur. The only way users can reset this condition is to reduce disk use below the specified quotas. The system manager can reset this condition by increasing the user's quota limits or turning quotas off.

Users modifying a file owned by someone else are subject to the quota limit of the owner of that file. When the soft limit is exceeded, the error message is issued to the owner of the file, and the hard quota time limit countdown begins for the owner of the file. Occupied disk space must be reduced below the limit to reset the condition.

This also applies to users extending files owned by root. If the user root quota limit is set to zero, which specifies no quota limits apply, all users with write permissions to files owned by root are allowed to write until the partition fills up, regardless of personal quota limits. Because of this, you must decide whether or not to place quota limits on the user root.

No warnings or error messages are printed if a user exceeds soft or hard limits on an NFS-mounted file system. If this is likely to cause problems at your site, instruct users running jobs on remote systems to run `quota` on the remote system before starting a job.

Checking data

The `quotacheck` command examines each file system, builds a table of current disk usage, and compares this table against that stored in the disk quota file for the file system. If inconsistencies are detected, both the quota file and the current system copy of the incorrect quotas are updated.

The system runs `quotacheck` automatically at boot time from the /sbin/init.d/hfsmount script.

You can also use `quotacheck` interactively.

Step 1 Log in as root.

Step 2 If it is not already mounted, mount the file system.

For example enter:

```
mount /dev/dsk/sd2b
```

Step 3 Check quotas with `quotacheck`. Enter:

quotacheck -v /home

```
*** Checking quotas for /dev/dsk/sd2b (/home)
/dev/dsk/sd2b: bob    fixed    files  12 -> 13   blocks  103 -> 128
```

The output indicates the quotas file for bob has been fixed or changed to reflect changes in the number of files and blocks. The `-v` option reports information for each user in the file system.

Managing SPP-UX file systems

4

SPP-UX V5.1 and V5.2 introduce a new file system layout. Affected areas include:

- File and directory locations—Covered in this chapter.
- System start up and shut down—Covered in "Starting and stopping SPP-UX," on page 1.

This chapter also covers day to day tasks involved in creating and managing SPP-UX file systems. Topics covered include:

- File system basics
- Creating file systems
- Mounting file systems
- Unmounting file systems
- Checking file systems
- Managing large file systems
- Managing available space

See the *HP-UX 10.0 File System Layout Whitepaper*, located on your system in /usr/share/doc/filesys.txt, for an explanation of the reasoning for, and impacts of, the new file system layout.

Terms

Terms used in this chapter include:

dynamic files

Files that change size and content during system usage. Examples include log files, temporary files, configuration data, and user files.

static files

Files that usually do not change. Examples include: binaries, libraries, and documentation.

Changes specific to this release

Many file and directory locations completely change with this release.

The `cnx_newfs` command is now located in /sbin.

Many file system administration commands are linked to `fs_wrapper`. `fs_wrapper` determines the type of file system the command is operating on and then invokes the file system-specific binary. `fs_wrapper` does not support large file systems operations; you must use the `cnx_newfs` command to create large file systems.

See the fs_wrapper(5) and cnx_newfs(1M) man pages for more information.

File reorganization

Static files—binaries, libraries, documentation, and headers—are separated from dynamic files—logs, temporary files, and configuration data.

Another layer separates the operating system from the application software. In previous releases, application software resided with the operating system software in /usr. This release moves application software to /opt.

For a thorough description of the contents of each directory see the *HP-UX 10.0 File System Layout Whitepaper,* located on your system, in /usr/share/doc/filesys.txt.

For a list of the current SPP-UX directories and a mapping to equivalent directories from previous SPP-UX releases see "SPP-UX directories," on page 153.

New start up and shut down model

The new model allows you to set and modify system behavior by changing script variables. Nonmodifiable execution scripts read the variables and start or stop subsystems according to arguments you supply. This model separates script execution and variables as shown in Table 8.

Table 8 Startup and shutdown directory locations

Script/file	Located in...	Description
Execution scripts	/sbin/init.d	Starts up and shuts down every subsystem. Execution scripts are not modifiable and will be over written in subsequent releases of SPP-UX
Configuration variable scripts	/etc/rc.config.d	Controls system behavior
Link files	/sbin/rc*.d	Controls the sequencing order of the execution scripts. Do not modify the ordering of these link files

If you have modified rc scripts to include customized functionality, you need to move the script information to comply with the new startup/shutdown model. For more information see the *HP-UX 10.0 File System Layout Whitepaper*, located on your system, in /usr/share/doc/filesys.txt.

For more information about changes to start up and shutdown processes, see "Starting and stopping SPP-UX," on page 1.

Man pages

Application manpages are located in the same /opt directory as their associated software.

Environment variables

Environment variables contained in system and user login files must be checked for nonconforming execution strings or path names.

Embedded pathnames

Hard coded pathnames may be found in:

- User's program sources, headers and make files
- User's scripts—`ksh`, `sh`, `awk`, `sed`, etc.

- Libraries
- Binaries and object code
- Message catalogs
- Help and man page files

Application file locations

Because each application is installed in a unique directory in /opt, users must keep their PATH and MANPATH environment variables up to date. To simplify this task, the following files are updated each time you install a new application:

/etc/PATH

Contains the default path for system files and application files. The default is /usr/bin, /usr/ccs/bin, and /usr/contrib/bin.

/etc/MANPATH

Contains the default path for system and application man pages. The default results in matches for all manpages found in /usr/share/man, /usr/contrib/man, and /usr/local/man.

File system basics

Exemplar Technical Server file systems include:

- High-performance File System (HFS)

  HFS file systems physically reside on mass storage devices, usually hard disk drives. HFS file systems provide the following support:

  - Short file names (14 byte maximum)
  - Long names (255 byte maximum)
  - Large files (Greater than 2GB)

    A large files capable file system is not mutually exclusive of short or long file name file systems.

- Network File Services (NFS)

  NFS file systems are remote HFS file systems, accessible over a network, that can be used in a local file system.

Creating a file system

After you create partitions and combine them into disk stripes as shown in "Managing disks," on page 37, you can construct a new file system with one the following commands:

`/sbin/newfs`

By default `newfs` creates file systems of the same type as the root file system.

`/sbin/cnx_newfs`

By default, the `cnx_newfs` command creates large file compatible file systems with block/frag sizes 64k/8k.

The `newfs` command is a user friendly front-end to `mkfs`.

Creating a root file system

When creating a root file system you must use one of the following commands:

- `newfs -L`
- `cnx_newfs -N -L`

These commands support OBP 's boot loader requirements for booting from a file system with the FS_MAGIC_LFN magic number. The following example creates a root HFS file system with long file names from the character special file `/dev/rdsk/sd0a` with the disk type `scalios`:

Step 1 Log in as root.

Step 2 Create the file system with newfs. Enter:

```
newfs -L /dev/rdsk/sd0a scalios
```

Creating a non-root file system

When you create a non-root file system, you must decide if you are going to create a regular file system or a large files file system. The file system type determines the command you will use:

- HFS file system—Use `newfs`

  The `newfs` command's default:

  - Block size is 8k
  - Frag size is 1k
- Large files file system—Use `cnx_newfs`

  The `cnx_newfs` command's default:

  - Block size is 64k
  - Frag size is 8k

See the newfs(1M) and cnx_newfs(1M) man pages for more information about available options and usage for these commands.

Mounting file systems

Once you have installed and mapped your disk and created a file system with `newfs` or `cnx_newfs`, you are ready to mount the new file system for use.

Mounting a local file system

To use the `mount` command to add a file system, complete the following steps:

Step 1 Log in as root.

Step 2 Determine the mount point directory for the file system.

Use `mkdir` to create the directory if it does not yet exist.

Step 3 Use the `mount` command to add the file system.

For example, you want to add the file system on a SCSI disk with the device file /dev/dsk/sd0c. The disk contains the files of most of the users of the system, so you specify the empty directory /users as the mount point directory for the file system. Enter:

```
mount /dev/dsk/sd0c /users
```

This command mounts the file system on the disk using the device file /dev/dsk/sd0c to the mount point /users.

Mounting file systems automatically

To automatically mount a file system at bootup, list it in the /etc/fstab file. See the fstab(4) man page for details on creating /etc/fstab entries.

Mounting a NFS file system

In this example, the /home file system on the remote host tempest is mounted in /home on the local host.

Step 1 Log in as root.

Step 2 Determine the name of the host machine and the file system's directory on the host machine.

Step 3 Test the network connection between the local machine and the host with `ping`. Enter:

```
ping tempest
```

Step 4 Ensure the /etc/exports file on tempest lists the /home file system as available to export.

Step 5 Export tempest's files with `exportfs`. On tempest, enter:

```
/usr/sbin/exportfs -a
```

Step 6 Mount the file system with `mount`. On your local host, enter:

```
mount -F nfs tempest:/home /home
```

Unmounting file systems

To unmount a file system, use the `umount` command. All files on the file system to be unmounted must be closed. Attempting to unmount a file system that has open files, including your working directory, causes the `umount` command to fail without unmounting the file system.

Unmounting a local file system

To unmount a local HFS file system on the disk with the device file /dev/dsk/sd0c mounted at the mount point directory users, issue the command:

```
umount /dev/dsk/sd0c
```

or

```
umount /users
```

Unmounting file systems automatically

When you execute the `shutdown` command, the system unmounts all of the file systems listed in the /etc/mnttab file. `shutdown` uses `umount -a` to unmount file systems. File systems are also unmounted when you use the `reboot` command.

Unmounting an NFS file system

To unmount the NFS file system /users exported from the remote host tempest, enter:

```
umount tempest:/users
```

Checking file systems

Check your file systems for consistency periodically and any time you suspect a problem. Some commands, such as `convertfs`, will not function properly unless the file system is free of inconsistencies.

Use the `fsck` utility to check file systems for any inconsistencies and to make any necessary repairs. When you run `fsck`, make sure the file systems are inactive.

Step 1 Make sure all users are logged off the system.

Step 2 Bring the system to single-user mode. Enter:

```
shutdown
```

This terminates running processes and places the system in single-user mode.

Step 3 Check your file system with `fsck`. Enter:

```
fsck -p
```

The `-p` option allows you to fix many file system problems, running non-interactively. See the fsck(1M) man page for information on `fsck`'s options. If `fsck` either finds no errors or finds correctable errors, it corrects any such errors and prints information about the file systems it checks.

If `fsck` encounters a problem it cannot correct while running with the `-p` option, it terminates with an error message. At this point, you must run `fsck` interactively.

In interactive mode, as `fsck` encounters errors, it requests permission to perform certain tasks. If you do not give `fsck` permission to perform the correction, it bypasses the operation, leaving the file system unrepaired.

Managing large file systems

SPP-UX large file systems allow the creation of files and file systems larger than 2 Gb. A large file system allows you to create and manipulate files up to one terabyte minus 512 bytes in length.

Caution

Do not create a large file system on the root partition. The OpenBoot utility can not boot SPP-UX on a root partition that contains a large file system. The root partition must be a standard HFS long-filename file system that is smaller than 2 Gb.

The OpenBoot file reader cannot follow symbolic links, so filenames must be hard links, like the standard /stand/spp3/mach, /stand/spp3/tunables, and /stand/spp3/server.

Using large file system utilities

The following file system utilities allow you to create and maintain large file systems:

`/sbin/cnx_dumpfs`

Prints the super block and cylinder group information for a file system. When used with the `-l` option, `cnx_dumpfs` prints only the file system magic number and feature bits.

If a file system's magic number is FD_MAGIC_2 and the feature bits are FSF_LARGEFILES, the file system allows the creation of files larger than 2 Gb.

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

`/sbin/cnx_mkfs`

Constructs a file system. This command uses a prototype specification for the file system; in order to create a large file system using `cnx_mkfs`, you must specify a prototype for a large file system.

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

`/sbin/cnx_newfs`

Creates a large file capable file system by default. The `cnx_newfs` command is a front end to `cnx_mkfs`.

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

Creating a large file system

To create a large file system, complete the following steps:

Step 1 Log in as root.

Step 2 Make sure the partition is unmounted.

Step 3 Create the large file system with `cnx_newfs`. Enter:

`/sbin/cnx_newfs` *special_file* **`scalios`**

where *special_file* is the character special file for the disk and `scalios` is the special disktype for Exemplar Technical Servers.

For example, to create a /dev/rdsk/sd1a partition capable of containing large files, enter:

```
/sbin/cnx_newfs /dev/rdsk/sd1a scalios
```

Checking a file system for large file capability

To check a file system in order to determine whether it is capable of having large files, use the `cnx_dumpfs` command. Enter:

`/usr/sbin/cnx_dumpfs -l` *filesystem*

where *filesystem* is the name of the file system you want to check.

The following output illustrates a file system that does not have large file capability:

```
/sbin/cnx_dumpfs -l /

magic    FS_MAGIC_LFN
featurebits     FSF_LFN
```

Your system's root partition (/) must not have large file capability.

The following output illustrates a file system that does have large file capability:

```
/sbin/cnx_dumpfs -l /scratch

magic    FD_MAGIC_2
featurebits     FSF_LARGEFILES
```

Managing available space

You should check free space on your files systems periodically to keep track of which ones are most active and tend to fill quickly. You can free space on heavily used file systems by advising users to clean up or by automating the process with the `cron` and `at` commands.

Using df and bdf

You can use `df` or `bdf` to list all mounted file systems and the amount of free space on each disk.

Output from `df` shows the number of available file system inodes. Note that by dividing the number of 512-byte blocks shown by two, you can get the available space in kilobytes, as reported by `bdf`.

These commands are located in /usr/bin.

Freeing disk space

The following suggestions help you free disk space when you notice a disk is filling up:

- Archive infrequently used files to tape or disk.

  Use `find -a` to check access time of files located on the problem disk. See the find(1) man page for more information.
- Prompt system users to remove obsolete files. Possibilities include:
  - Core files
  - Log files
  - Exceptionally large files
- Move files from a heavily used disk to a less active disk.

  Files in the following directories should not be moved: / (root), /sbin, /stand, /etc, /dev, and /usr.

Using the Software Distributor 5

The Software Distributor (SD) lists, configures, and verifies installed software and allows multiple version and source locations of software. Use SD to manage and track software installed on your Exemplar Technical Server.

This chapter:

- Gives an overview of SD
- Lists SD commands and briefly describes their functions.

For a more in-depth description of SD, see *Managing software with SD-UX* (B2355-90080), the sd(8) man page, and man pages for SD commands listed in "SD-UX commands" on page 71.

Terms

Terms used in this chapter include:

SD (Software Distributor)

Management tool for installing, listing, configuring, and removing software.

agent

Performs basic software management tasks.

controller

Provides a user interface for management tasks, collects and validates data, displays task status information, and invokes SD-UX commands.

depot

Gathering place for software to be installed by SD. Can be a directory location or host that contains software packaged in SD format.

filesets

The smallest manageable SD software unit, a collection of files.

host

Performs SD-UX operations.

product

A complete software product. Can contain subproducts, filesets, and files.

subproduct

An optional grouping of filesets used to partition a product that contains many filesets, or to offer the user different views of the filesets.

target host

Install destination. Can be a computer or a directory.

SD overview

SD works in a networked environment to help you manage and install software. Daemon processes schedule and perform management tasks on related software—software objects—by creating lock files and assigning a succession of "states" or conditions to the software objects. A database, the Installed Product Database (IPD), tracks the status of each software object under SD's management.

SD daemon/agent

Every target host and depot host must have the following two processes when using SD:

- `swagentd`—SD daemon; listens for controller requests and schedules the agent process —`swagent`— to perform tasks.
- `swagent`—SD agent; performs management tasks such as software installation, configuration, and verification.

The distinction between these processes is invisible; they are often referred to as a single process, the SD daemon/agent.

The daemon process must be scheduled before your system is available as a target host or source system. You can do this manually or in your system start-up script. `swagentd` executes `swagent` to perform software management tasks; never execute `swagent` by hand.

SD software objects

Software organization consists of three levels—from small filesets to entire products. Each level is a software object, and selecting a software object is called software selection.

Table 9 defines SD software objects.

Table 9 SD software objects

Software object	Definition
Fileset	Smallest selectable software unit. Includes files and control scripts of a single product
Subproduct	Collection of logically grouped filesets. Used if a product has many filesets
Products	Contains all components of a product; can be subproducts, filesets, and files
Bundle	Contains a set of products

SD software states

SD commands move products and filesets through a number of states. Table 10 lists software states.

Table 10 SD software states

State	Description
CONFIGURED	Installed and ready to use
AVAILABLE	Accessible to install by target host
INSTALLED	Present on target host, not ready to use
TRANSIENT	Installation or removal in process
CORRUPT	Failed move during TRANSIENT state
NOT FOUND	Not known to SD (non-existent)

During installation software moves through the following states:

1. Nonexistent
2. TRANSIENT
3. INSTALLED
4. CONFIGURED

During removal, software moves through the following states:

1. CONFIGURED
2. INSTALLED
3. TRANSIENT
4. Nonexistent

When you package or copy software into a depot, the software moves through the following states:

1. Nonexistent
2. TRANSIENT
3. AVAILABLE

When you remove software from a depot the software moves through the following states:

1. AVAILABLE
2. TRANSIENT
3. Nonexistent

Installed products database (IPD)

Upon installation, SD writes each software object's product information or attributes to a collection of files or subdirectories located under root (/). Attributes can include:

- Product name
- Files and filesets
- Part numbers
- Size
- Vendor
- Revision

IPD's default location is /var/adm/sw/products. You can see a product's attributes by using the `swlist` command.

Do not manually change this information: you can corrupt the IPD and cause SD commands to fail. Use SD commands to view and modify the database. Their functionality automatically updates the database.

SD also maintains depot catalogs, on the SD host machine, that contain information about all products available in the depot with the same structure and type of information as the IPD. By default, depot catalogs are located in /var/spool/sw/catalog.

SD-UX commands

The contents of an SD depot reside on the host in a directory structure with a single, common root. A depot can exist as a directory tree on a SPP-UX file system or it can exist as a tar archive on a serial media. All depots share a single logical format, independent of the type of media on which the depot resides. This section lists and briefly describes SD commands. These commands are located in the /usr/sbin directory.

`swinstall`

Installs and configures software products.

`swcopy`

Copies software products for subsequent installation or distribution.

`swremove`

Unconfigures and removes software products.

`swlist`

Displays information about software products.

`swconfig`

Configures, unconfigures, or reconfigures installed software.

`swverify`

Verifies software products for compatibility and completeness.

`swpackage`

Packages software products into a distribution directory or distribution tape.

See the related man pages for more information on these SD commands.

Configuring processor and memory resources

6

To help control and use your Exemplar S-Class and X-Class Technical Server's processors and memory resources, the operating system provides a configuration utility, the Subcomplex Manager.

The Subcomplex Manager provides both a graphical user interface (GUI) and a command-line interface. The command-line interface is useful to:

- Configure the system following reboot as part of the startup procedure
- Reconfigure the system from script files between jobs in a batch queue
- Load subcomplexes
- Remove subcomplexes
- Reconfigure code server sets and global memory

For more information see "SCM command-line interface" on page 86 and the scm(1) man page.

The GUI enables the system administrator to maintain, load, and modify the subcomplex database. For more information see "Subcomplex Manager GUI" on page 75 and "Working with configuration files" on page 82.

You can perform the same functions with both the command-line interface and the GUI.

Terms

Terms used in this chapter include:

Subcomplex Manager

Allows you to maintain a database of subcomplex definitions, load subcomplex definitions onto the system, and modify subcomplexes currently running on the system.

complex

Is the complete set of processor and memory resources available on your Exemplar Technical Server.

hypernode

A set of processors and physical memory organized as a symmetric multiprocessor (SMP) running a single image of the operating system microkernel. An SPP system consists of one or more nodes, with a high speed CTI connecting the nodes.

subcomplex

Is the primary unit for allocating computing resources to users. A subcomplex is a collection of processors and global memory from one or more nodes of the Exemplar Technical Server.

system subcomplex

Is a special subcomplex, created automatically at boot time, to run system processes.

Changes specific to this release

The `scm` command is located in the /usr/sbin directory.

Subcomplex overview

The primary unit for allocating computing resources to users is the subcomplex. A subcomplex is a collection of processors and global memory from one or more nodes of the system. Any number of processors from any number of hypernodes may be allocated to the subcomplex. Global memory may be allocated on any hypernode having at least one processor assigned to the subcomplex. Memory is allocated per hypernode per subcomplex.

Every system user may be 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.

System subcomplex

The system subcomplex is a special subcomplex, created automatically at boot time, to run system processes. The system subcomplex often also runs user jobs, especially login shells.

You can modify the system subcomplex, but leave a minimum of two processors on the root node (node 0) allocated to the system subcomplex.

The Subcomplex Manager does not allow you to remove the last processor from the system subcomplex.

Allocating more processors to the system subcomplex increases the percentage of the system's computing resources devoted to operating system functions-such as initd, init, and rlogind.

Allocating fewer processors to the system subcomplex increases the number of processors available for other subcomplexes.

Monitor the performance on your system subcomplex with the `syspic` utility to help you determine the best split of system and user resources.

See `syspic`'s online help and "Monitoring system performance" on page 91 for more information.

Subcomplex Manager GUI

The Subcomplex Manager GUI, invoked with the `scm` command, enables you to specify and load hypernode subcomplex configuration file system resources and to create complex configuration files.

The Subcomplex Manager GUI consists of the Subcomplex Manager main window, the Node Attributes dialog box, and the Subcomplex Attributes dialog box.

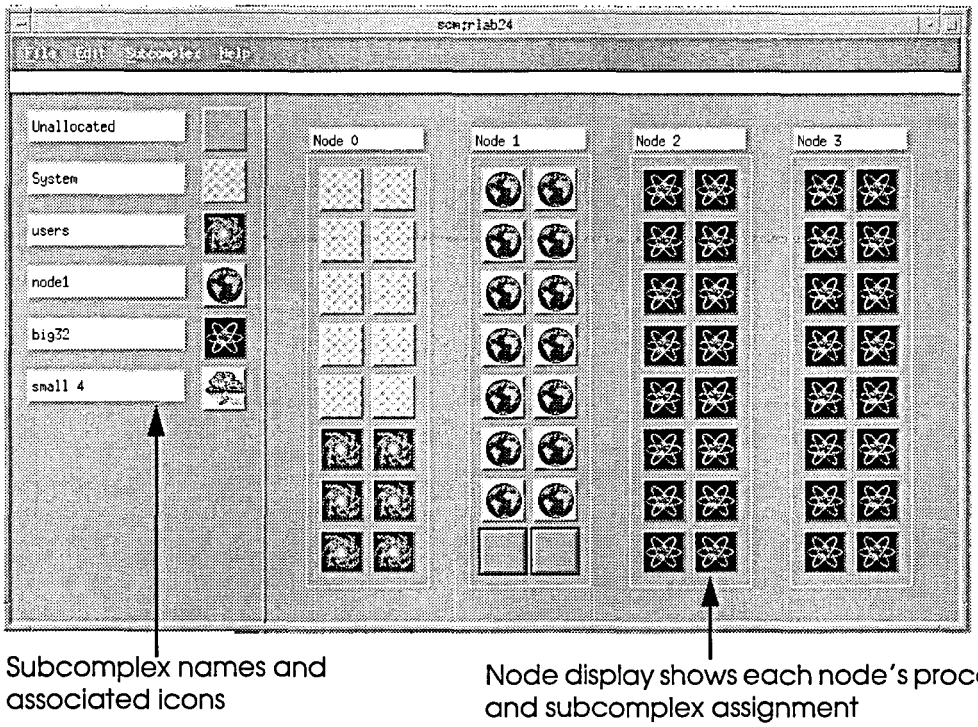


Figure 2 Subcomplex Manager main window

Main window

The main window, as shown in Figure 2, displays a list of subcomplexes and a representation of all nodes and processors in the system. This window displays the current complex configuration when you start the interface unless you specified the `-n` option in the `scm` 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 subcomplex list, the node display, and the menu bar.

Subcomplex list

The subcomplex list appears on the left side of the window by default and lists subcomplexes 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:

- View and edit information about a subcomplex—Select a subcomplex name to activate the Subcomplex Attributes dialog box.

- Assign all chosen processors to a subcomplex—Select a subcomplex icon to assign all processors currently chosen in the Node display to that subcomplex.

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:

- View and edit detailed information about the node

  Select a node name to activate the Node Attributes dialog box.
- Assign a set of processors to a subcomplex

  Highlight a set of processors and select the subcomplex's icon from the Subcomplex list.
- Assign a set of processors to a nodes' server set, if the Server set list is currently displayed in the left side of the window
- Remove a set of processors from a node's server set, if the Server set list is currently displayed in the left side of the window

Select a single processor by left-clicking on it. Select a contiguous group of processors by 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 remain selected.

Dialog boxes

Sample dialog boxes—the Subcomplex Attributes dialog and Node Attributes dialog—are shown in Figure 3.

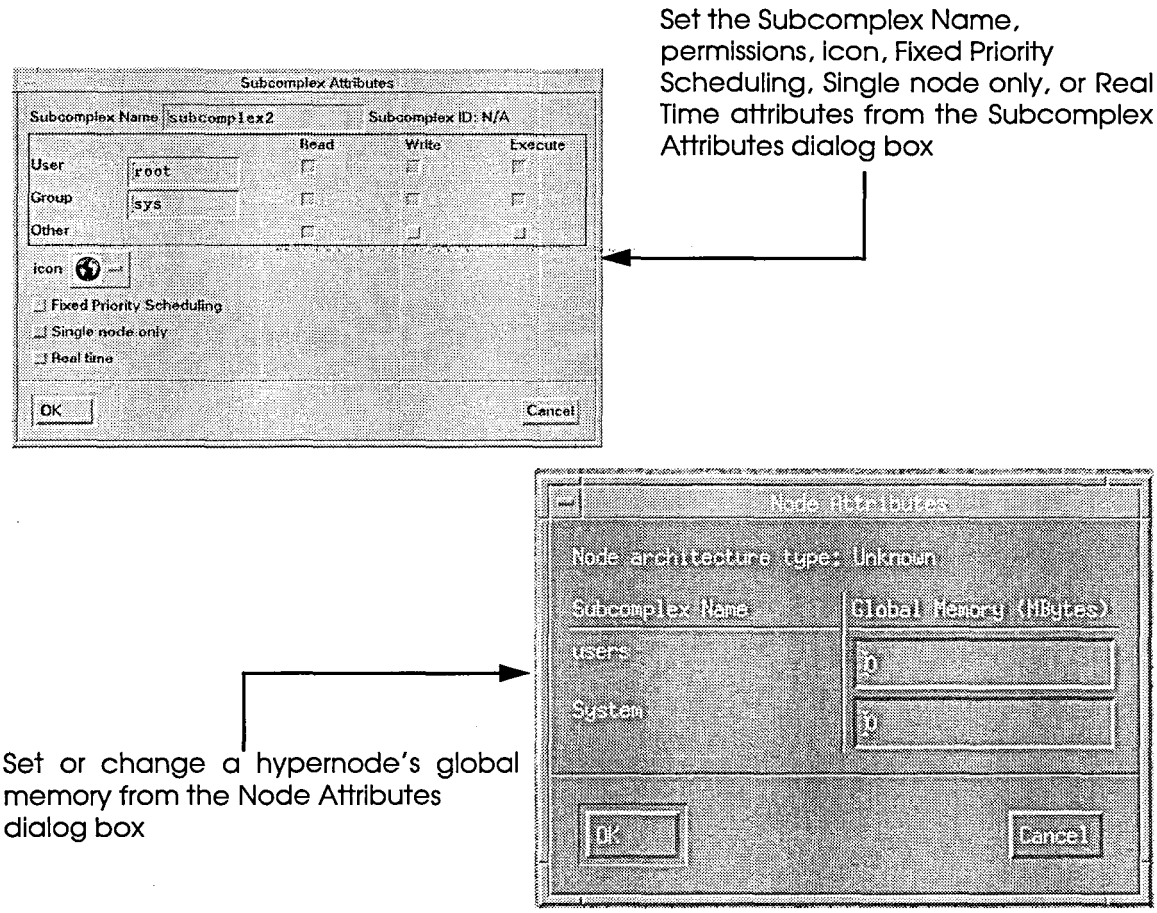


Figure 3 Subcomplex Manager dialogs

Node Attributes dialog box

To set or change information about a particular hypernode's assigned global memory, left click on the hypernode name in the Node display (right side of the main window.) The dialog shows each node's:

- Subcomplex names
- Designated global memory, in Mbytes, for all associated subcomplexes

To allocate global memory, enter the amount of the node's global memory—in 16 Mb increments— you want to allocate to each listed subcomplex. When you have finished changing values, select the OK button to make the changes or select the Cancel button to cancel the changes and exit the dialog.

Subcomplex Attributes dialog box

To view detailed information about a particular subcomplex, left click on the name of that subcomplex in the Subcomplex list (left side of the main window.)

Selecting a subcomplex from the subcomplex list portion of the Subcomplex Manager window pops up the Subcomplex Attributes dialog box. The dialog displays information about a single subcomplex, including:

- Subcomplex name
- Subcomplex ID
- Ownership
- Read, write, and execute permissions
- Scheduling policy
- 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

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

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

Permissions

Read, write, and execute permissions for the user, group, and world, or other. Read permission indicates that the user or group may view the subcomplex definition. Write permission indicates that the user or group may modify the set of enabled scheduling policies for the subcomplex. Execute permission indicates that the user or group may run processes on the subcomplex.

Icon

Click on the icon shown to view a list 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.

Scheduling policies

Time Share, the default, is always enabled on the subcomplex. Select this field to enable fixed priority scheduling.

Single node only

Restricts the subcomplex to a single node which allows the operating system to use local memory when an application asks for global memory. A subcomplex with this attribute can not have GMEM configured.

Real time

Enables real time functionality. This functionality requires a license.

Configuration tips

Some constraints apply when you first configure a subcomplex, when you reconfigure the complex while in use, or when making changes to a subcomplex currently loaded on the complex—even if it is not yet in use. A subcomplex with one or more user processes currently assigned to it is "busy."

Resource allocation restrictions

The following restrictions apply when you initially allocate system resources:

- Do not exceed the total memory available on the hypernode when you allocate global memory.
- Balance the use of memory and processors between system and user subcomplexes:
  - To increase the percentage of resources for operating system functions—Allocate more than one processor to the system subcomplex.
  - To increase the percentage of resources available for user programs—Allocate fewer processors to the system subcomplex; but, leave a minimum of two processors on the root node (node 0) allocated to the system subcomplex.

Reconfiguration restrictions

The following restrictions apply when you reconfigure a subcomplex while it is busy or when you make changes to a subcomplex that is currently loaded in the complex (even if it is not in use):

- To remove a busy subcomplex—You must kill all user processes on that subcomplex. The Subcomplex Manager does not kill processes; you must do this explicitly using the `kill` utility. To fulfill the `kill` command, the subcomplex must have at least one processor allocated.
- To reconfigure global memory—You cannot change the global memory of a subcomplex once any global memory has been allocated. You must kill all user processes on that subcomplex, delete the subcomplex, recreate it, and reboot the machine.
- If a subcomplex already has global memory allocated, you cannot add new processors *unless* they belong to one of the nodes that is already being used by the subcomplex. This constraint applies even if the subcomplex is not currently busy.
- You can not expand a subcomplex to an additional node if global memory is configured.
- You can not configure global memory on a subcomplex with the Single node only attribute enabled.
- You can not configure global memory for a single node machine.

System subcomplex reconfiguration restrictions

The following restrictions apply to the system subcomplex:

- The system subcomplex can never be removed from the complex.
- Leave at least two processors on the root hypernode (node 0) assigned to the system subcomplex.

The Subcomplex Manager does not allow you to destroy or remove the last processor from the system subcomplex.

Allowable subcomplex reconfiguration actions

The following actions are allowed at any time on a busy subcomplex:

- You can add processors to a subcomplex.
- You can remove processors from a subcomplex.

While not recommended, you can remove all processors for a subcomplex from a node. Doing so suspends all threads running in that subcomplex.

- You can change the following subcomplex attributes:
  - Name
  - UID
  - GID
  - Permissions
- You can enable or disable scheduling policies.

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.

Working with configuration files

This sections describes:

- Starting the Subcomplex Manager
- Creating a subcomplex
- Changing a subcomplex
- Removing a subcomplex

Starting the SCM GUI

The Subcomplex Manager GUI requires an X Windows server running X Windows Version X11R5. To start the Subcomplex Manager GUI, follow these steps:

Step 1 Set your `DISPLAY` environment variable to the X Windows server you are using. For example:

```
setenv DISPLAY mydisplay:0.0
```

where **`mydisplay`** is your display name.

Step 2 Enter the `scm` command with no command-line options:

```
scm
```

The Subcomplex Manager's main window appears.

Creating a subcomplex

To create a new subcomplex complete the following steps:

Step 1 Start the Subcomplex Manager. Enter:

```
scm
```

Step 2 Create the subcomplex. Select Create from the Subcomplex menu.

Designate the Subcomplex Name, set permissions for User, Group, and Other users and choose the icon you want to represent the subcomplex. If appropriate, choose Fixed Priority Scheduling and Single node only. See "Subcomplex Attributes dialog box" on page 79 for more information about these attributes.

Step 3 Assign the processors you wish to use for your new subcomplex.

Select a single processor by left-clicking on it. Select a contiguous group of processors by 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 remain selected.

Step 4 Assign the processor to the new subcomplex by clicking on the subcomplex's associated icon.

Adding or changing a subcomplex

Once you have created a satisfactory subcomplex configuration you can apply it to your system by completing the following steps:

Step 1 Configure the subcomplex you want in the Subcomplex Manager window.

Step 2 Apply your changes. Select Perform Reconfiguration from the File menu. Confirm your changes by selecting OK on the Reconfiguration confirmation dialog.

The Subcomplex Manager returns one of the following:

- A summary of changes
- A list of subcomplexes that must be idled before the new configuration can be applied

  This list appears only if you try to remove all processors for a subcomplex from a node; in this case, you must kill all processes running in that subcomplex.
- The reasons a configuration cannot be applied to your system

Deleting a subcomplex

Before you delete a subcomplex check to make sure you:

- Have killed all user processes
- Are not removing the last processor from the system subcomplex

To delete a subcomplex complete the following steps:

Step 1 Choose Delete from the Subcomplex menu.

Step 2 Highlight the name of the subcomplex you wish to delete from the resulting Delete subcomplex dialog and choose OK.

SCM configuration file format

The SCM GUI automatically creates an SCM configuration file. See the scm(4) man page for a sample configuration file. This file can be stored anywhere in the file system. You can also create or edit the file manually.

The configuration file consists of a list of subcomplex definitions. All 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` keyword, *value* must be an integer.

The SCM configuration file can contain the following keywords:

`SC`

The name of the subcomplex. All statements following this statement, up to the next SC statement, are part of this subcomplex's 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 subcomplex permissions, determines the actions the owner performs 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 GID of the group that uses the subcomplex. This value, along with subcomplex permissions, determines the actions the group performs on the subcomplex. Generally, this group has permission to execute processes on the subcomplex or execute permission. The default is the effective GID of the user who invokes the `scm` command.

PERM

The read, write, and execute permissions—similar to file system permissions—for the subcomplex owner, user group, and the world. 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

Denotes the scheduling policy for the subcomplex, either fixed-priority scheduling or timeshare scheduling. The values for this keyword are defined in /usr/include/sys/cnx_tattr.h

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.

GMEM

The amount, in 16 Mb increments, of global memory allocated to this subcomplex on the node specified in the previous NODEID entry. The default value is 0.

SINGLE_NODE_ONLY

Restricts the subcomplex to a single node. This allows the operating system to use local memory when an application asks for global memory. A subcomplex with this attribute enabled can not have GMEM configured. A value of 1, the default, restricts subcomplex memory access to a single node, a value of 0 uses global memory.

REAL_TIME

Enables real time functionality when set to 2. This functionality requires a license.

SCM command-line interface

You activate the GUI if you specify no options or the `-n` option to the `scm` command. You can perform the same functions with both the command-line interface and the GUI.

If you specify any other option, the command runs without the GUI. The `scm` command has the syntax:

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

where

`-c`

Returns a full description of the current system configuration. If you are not running with superuser privileges, only information about subcomplexes for which you have read access is returned.

`-l` *filename*

Loads the complex configuration from *filename* to the system. Requires superuser privileges.

`-n` *nnodes*

Creates a hypothetical machine, with *nnodes* nodes, and performs all specified operations without affecting the actual configuration. Useful for attempting trial configurations. This option disables the `-l`, `-o`, and `-r` options and their equivalent functions in the GUI.

`-o` *filename*

Overlays the complex configuration from *filename* to the system. Requires superuser privileges.

`-r` *scname*

Removes subcomplex *scname* from the system. Requires superuser privileges.

`-s`

Returns a list of the names of all subcomplexes currently loaded on the system. Only those complexes for which you have read access are displayed.

Assigning jobs to subcomplexes

Use the `mpa` command to run a program within a specific subcomplex. Usage is:

`mpa [-sc` *sc_name* `|-scid` *sc_id*`]` *filename*

where

sc_name

Designates the name of the subcomplex on which to run the program *filename*.

sc_id

Designates the subcomplex ID number on which to run the program *filename*.

The `mpa` command has many options beside those listed here, see the mpa(1) man page for more information.

Monitoring performance 7

SPP-UX provides several utilities to monitor the performance of your Exemplar S-Class and X-Class Technical Server:

- `sysinfo`
- `syspic`
- `ps` and `cnx_ps`
- `top` and `pot`

Printing system information

The `sysinfo` utility gets its information from the `cnx_sysinfo` system call. Command syntax and descriptions are listed in Table 11.

Table 11 Command options for `sysinfo`

Option	Description
`-a`	Prints all available system information
`-mv`	Prints the Mach kernel version string
`-sv`	Prints the SPP-UX server version string
`-av`	Prints the AIL library version string
`-cpu` *phys_node cpu*	Prints basic CPU information. With no additional parameters, information for all nodes in the system is printed. Given a physical node ID, output is limited to that node. Given both a physical node ID and a CPU ID, the output is for the one CPU only
`-cpu_count`	Prints the total number of CPUs in the entire system
`-cpu_type`	Prints the architecture type of the system. For example: SPP1200
`-lc`	Prints the mean load average of the entire system
`-lcn` *phys_node*	Prints the load average for each node in the system, or with a physical node argument, prints the load average for that node

Table 11 Command options for `sysinfo` —**(continued)**

Option	Description
`-ls` *sc_name*	Prints the load average for all subcomplexes, or for a particular subcomplex, if given a subcomplex name as an argument
`-lsn` *sc_name logical_node*	Prints the load average for all nodes of all subcomplexes. Given a subcomplex name, prints all nodes for that subcomplex. Given both a subcomplex name and a logical node id for that subcomplex, prints only the load average for the specified node in the specified subcomplex
`-memc`	Prints memory statistics for the entire system. Prints the total, allocated, and free memory, in Mb, for all node private, global, network cache, and buffer cache memory in the system
`-memn` *phys_node*	For all nodes in the system, prints the total, allocated, and free memory, in Mb, for all node private, global, network cache, and buffer cache memory on that node
`-msg`	Prints the System V message parameters: `msgmax`, `msgmnb`, `msgmni`, and `msgtql`
`-i`	Prints the machine's identification number
`-node` *phys_node*	Prints basic physical node information for each node in the system, or for one node if given a physical node ID. Output shows the total memory in megabytes, the number of CPUs, and the list of CPU ID's
`-sema`	Prints the System V semaphore parameters: `snmmni`, `smmmsl`, `semopn`, `semume`, `semvmx`, `semaem`
`-shmem`	Prints the System V shared memory parameters: `shmmax`, `shmmin`, `shmmni`, `shmseg`
`-stat`	Prints the SPP-UX server statistics, `syscall_cnt` and `fork_cnt`

Monitoring system performance

Use the `syspic` utility to monitor your Exemplar Technical Server's performance.

The Graphical User Interface (GUI) displays statistics for each of the following modules:

- Utilization
- Buffer cache
- Context switches
- Exceptions
- Interrupts
- Load/CPU
- Memory
- Network
- Paging

Some module information is grouped by hypernode or subcomplex.

The `syspic` utility can launch other programs to collect the information it needs. See the online help available from the GUI for an overview of `syspic` functions and statistics.

`syspic` accepts all standard X Toolkit command-line options as well as those listed in Table 12.

Table 12 Command line options for `syspic`

Option	Description
`-f` *fname*	Loads data from *fname*
`-r` *fname*	Records data to *fname*
-period	Sample time period in seconds
`-stbc`	Gathers buffer cache statistics
`-stcs`	Gathers context switch statistics
`-ste`	Gathers exception statistics
`-sti`	Gathers interrupt statistics
`-stl`	Gathers load/CPU statistics
`-stm`	Gathers memory statistics
`-stn`	Gathers network statistics
`-stp`	Gathers paging statistics
`-stu`	Gathers totalization statistics

Reporting process status

Check process status with the `ps` and `cnx_ps` commands.

`ps` invoked with no options prints information about active processes associated with the current terminal. The output consists of a short listing and contains only the process ID, terminal identifier, cumulative execution time, and the command name. See the ps(1) man page for a list of valid options.

Under SPP-UX, `cnx_ps` can obtain thread information about selected processes when you specify the `-T` option. Options using lists as arguments can have the list specified in one of two forms:

- A list of identifiers separated from one another by a comma
- A list of identifiers enclosed in double quotes and separated from one another by a comma or one or more spaces.

The `cnx_ps` command accepts the same command-line options as `ps` in addition to those listed in Table 13.

Table 13 Command line options for `cnx_ps`

Option	Description
`-T`	Provides thread information
`-s` *scID*	Shows processes from subcomplex identified by *scID*
`-s` *scname*	Shows processes from subcomplex identified by *scname*

Table 14 describes the command's output fields.

Table 14 `cnx_ps -T` output fields

Field	Description
PID	Process ID
UID	Owner of the process
TID	Thread ID
S	State
USRTIME	User run time for the thread (min:sec)
SYSTIME	System run time for the thread (min:sec)
C	Processor utilization for scheduling

Table 14 `cnx_ps -T` output fields —**(continued)**

Field	Description
PRI	Processes priority, higher numbers mean lower priority
SLPTIME	Time a thread has been sleeping (min:sec)
SCID	Subcomplex ID
COMM	Command executed

Checking processes

The `top` command displays information about the processes on the system and periodically updates the information.

The `pot` command is a thread based `top`. `pot` extends the functionality of `top` by allowing you to specify the information to display and the order in which to sort that information. See the man pages for pot(1) and top(1) for more information.

Table 15 lists some of the valid `pot` command-line options. See the pot(1) man page for a complete list.

Table 15 Command line options for `pot`

Options	Description
`-F` *list*	List, similar to `cnx_ps` output, of fields to print
`-O` *list*	List of fields for sort order
`-d` *count*	Update display *count* times and exit
`-s` *time*	Screen update delay in seconds. Default is 5 seconds
`-p` *pid*	Display information for process *pid*
`-E` *expression*	Display only data that fits *expression*
`-e`	Lists all processes
`-f`	Output is similar to `ps -f`
`-l`	Output is similar to `ps -l`
`-u` *username*	List processes belonging to *username*

Managing printers

8

Once you add a printer to your system, you can add it to the line printer spooling system (lpss). In a networked environment if you do not add your printer to the lpss, there is no coordination between multiple users and intermixed listings can occur. This chapter covers management of:

- Line Printer Spooling System (lpss)
- Printers
- Print requests

This chapter describes how to accomplish the tasks associated with these topics using SPP-UX commands.

You cannot use `sam`, Hewlett-Packard's System Administration Manager utility, to setup or maintain user accounts or groups.

Terms

Terms used in this chapter include:

interface script

A shell script, located in the /var/spool/lp/interface directory, that is the final stage of the lpss print subsystem. 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.

line printer scheduler

The central program that must be running to ensure coordination of requests from users to printers.

local printer

A printer, configured into your lpss print subsystem, that is physically connected to your computer. Local printers are not supported on Exemplar S-Class and X-Class Technical Servers. See *remote printer*.

print destination

A print destination is a generic term used to describe a printer or plotter.

print queue

Also known as request directories, print queues are directories used by the lpss print subsystem to hold print jobs for each print destination until they can be printed.

print request

A term used to refer to a specific print job in the lpss print subsystem.

print request identification number

The number the lpss print subsystem uses to identify your print request. This identification number consists of the name of the printer followed by a sequence number.

printer interfaces

Responsible for sending data to a printer. Each printer you define for use by lpss has its own interface script that resides in the /var/spool/lp/interface directory. When printers are added to lpss, an interface script is copied from /var/spool/lp/model to /var/spool/lp/interface and given the printer name.

printer models

ASCII files that identify printer characteristics-such as whether your printer uses Printer Command Language (PCL) or PostScript. When you configure your printer into lpss, you must specify which printer model interface script you want to use.

printer names

A name that you refer to when you submit print requests. Printer names can contain up to 14 characters, which can be alphanumeric or underscores.

priority

A value associated with each printer and print request. Priorities control which print requests can print on a given printer. Priorities can be adjusted and must have a range from 0 to 7. Also known as fence priority.

remote print requests

A print request issued via the lpss print subsystem on your system to be printed on a printer attached to a remote computer.

remote printer

A printer in your lpss print subsystem but physically connected to another computer and accessed over a network. To configure a remote printer into your local lpss, you must be able to access the remote system via a LAN.

remote spooling

The process used to allow printing to printers defined as part of your lpss print subsystem but physically connected to another computer.

system default printer

Receives any print requests not sent to a specific print destination. You appoint one of the print destinations in your lpss to be the system default printer.

Changes specific to this release

The following commands reside in the /usr/sbin directory:

- `accept`
- `lpadmin`
- `lpsched`
- `lpshut`
- `lpmove`
- `lpmove`
- `lpfence`
- `reject`

You cannot use `sam`, Hewlett-Packard's System Administration Manager utility, to setup or maintain user accounts or groups.

Starting and stopping lpss

To control the flow of data through your printing system:

- Start or stop the LP spooler. This has a global effect. If you stop the LP spooler, all printing stops.
- Accept or reject any new print requests for a printer.
- Enable or disable a printer for printing.

To print, a printer must be accepting and enabled.

Overview

Think of lpss as a plumbing system. The data to be printed represents the water in this system. Various request directories, sometimes referred to as printer queues, 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 lpss as valves would control the flow of water in a plumbing system. Shell scripts (called interface scripts) near the end of the data flow serve as pumps that move 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.

If the "drain gets clogged" for one printer, you can reroute the print requests for that printer to another printer and you can "flush" unwanted print requests from the spooling system. You can also send a print request to another computer to be 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 called `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.

Starting lpss

Initial LP spooler setup consists of the following tasks:

- Add at least one printer to lpss.
- Tell lpss to accept print requests for this printer.
- Tell lpss to enable the printer for printing.
- Start the LP spooler.

See the "Adding printers" on page 103 for adding printers and accepting and enabling print requests.

To start lpss complete the following steps:

Step 1 Ensure that you have superuser capabilities.

Step 2 Make sure lpss is stopped. Enter:

```
lpstat -r
```

The LP system prints one of the following messages:

- `scheduler is running`
- `scheduler is not running`

If the scheduler is not running complete Step 3.

Step 3 Start lpss. Enter:

```
lpsched
```

When lpss starts, requests in the print queue completely reprint, regardless of their status prior to lpss shutdown.

Logging and analyzing printer activity

Analyzing printer activity helps you determine if there are bottlenecks in lpss. It also helps you determine and justify the need for additional printers. There are facilities to help you analyze the flow of data through your lpss.

There are two phases to analyzing lpss activity:

- Data collection phase
- Data reporting phase

The data collection phase begins when lpss starts. The `-a` option to the `lpsched` command turns on data collection processes when you start the LP spooler. The data reporting phase occurs any time after lpss starts. 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

Displaying printer statistics

To display statistics about printer activity, use the `lpana` command. The `lpana` command has the format:

`lpana` [-d*dest*]

where *dest* defines the printer for which statistics are displayed.

By default, `lpana` reports statistics for all printers.

Enabling activity logging

Start lpss with the following command to create a log of activity in the /var/adm/lp/lpana.log file:

```
/usr/sbin/lpsched -a
```

Stopping lpss

Before you stop the line printer scheduler, beware of the following:

- All requests currently printing are stopped, but remain in the print queues until you restart the scheduler
- Any print requests currently printing are completely reprinted when you restart the scheduler. This includes the print requests that were printing page 9,999 of a 10,000 page printout

To stop lpss, complete the following steps:

Step 1 Ensure that you have superuser capabilities.

Step 2 Check the status of local printing requests. Enter:

```
/usr/bin/lpstat -i -o
```

Stop the LP spooler when there are no requests currently printing.

Step 3 Stop lpss. Enter:

```
/usr/sbin/lpshut
```

Step 4 Restart lpss. Enter:

```
/usr/sbin/lpsched
```

Managing printers

Once you have started your lpss you need to add printers. This section describes the LPDEST environment variable, the rlpdaemon, and details the following tasks:

- Adding printers to the lpss
- Removing printers from the lpss
- Enabling printers
- Disabling printers

Adding a printer to lpss is not the same 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 (lpss)—that manages printer output.

LPDEST environment variable

To assign a default printer, set the `LPDEST` shell environment variable to the name of the system default printer. When users submit print requests:

- If `LPDEST` is set and a user does not specify a different printer to use, the printer referenced by `LPDEST` is 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—is used
- If neither `LPDEST` nor the system default printer is set, a user must specify a printer

Remote printers and `rlpdaemon`

If you have several systems connected to a Local Area Network (LAN) and would like the systems to share a printer, you can set up lpss on each system not physically connected to the printer. This automatically sends print requests—via the LAN— to the lpss of the system that does have the printer.

Systems without printers act as users on the system with the printer, submitting print requests to the remote 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 remote print requests from other systems. When these requests arrive, the daemon submits them to its local lpss on behalf of the remote user.

Remote cancel and status requests

In addition to remote print requests, the remote spooling daemon must also handle cancel and status requests from remote systems.

Special interface scripts on the remote systems issue cancel and status requests. These special interface scripts are similar to printer interface scripts. They have a model directory that holds sample versions of the scripts, and they have an interface directory where scripts currently in use reside. The cancel and status models are copied into their respective interface directories automatically when you add a remote printer. Table 16 describes remote cancel and status interface scripts.

Table 16 Remote cancel and status interface script descriptions

Location	Interface script	Purpose
/var/spool/lp/cmodel	`rcmodel`	Sends a remote cancel command to the system with the printer. When you configure a remote printer into your lpss, this script is copied into the /var/spool/lp/cinterface directory with the same name as the printer.
/var/spool/lp/smodel	`rsmodel`	Sends a remote status command to the system with the printer. When you configure a remote printer into lpss, this script is copied into the /var/spool/lp/sinterface directory with 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 scripts, copy them to a different file name to avoid destroying your changes when you update the system.

Adding printers

Exemplar Technical Servers do not support local printers. This determines several options for the `lpadmin` command you use when adding a printer. Use the `lpadmin` options detailed in Table 17 when you add a printer to your lpss.

Table 17 Predetermined `lpadmin` options

lpadmin flag	Description	Option
`-v`	Device file	`/dev/null`
`-ocm`	Cancel interface script	`rcmodel`
`-osm`	Status interface script	`rsmodel`
`-m`	Model interface script	`rmodel`

In addition to the information in Table 17, you need to gather the information detailed in Table 18.

Table 18 Additional `lpadmin` options

lpadmin flags	Description
`-orm`	Name of the remote system with the printer
`-orp`	Name of the destination printer—as defined in lpss of the remote system
`-orc`	(Optional)Designates users may only cancel their own print jobs
`-ob3`	(Optional)Designates the remote printer uses a BSD type printer daemon

See the lpadmin(1M) man page for additional options.

The following example adds a printer, slap, hosted from fried:

Step 1 Ensure that you have superuser capabilities.

Step 2 Stop the LP spooler with the `lpshut` command:

```
lpshut
```

Stop the LP spooler when there are no requests currently printing.

Step 3 Add the printer. Enter:

```
/usr/sbin/lpadmin -pslap -v/dev/null -mrmodel -ormfried -orpslap
-ocmrcmodel -osmrsmodel -orc -ob3
```

When you use the `lpadmin` command, do not put any spaces between the options and their respective values.

Step 4 Allow print requests. Enter:

```
/usr/sbin/accept slap
```

Step 5 Enable the newly added remote printer. Enter:

```
/usr/bin/enable slap
```

Step 6 Start the line printer scheduler with the `lpsched` command:

```
/usr/sbin/lpsched
```

Removing printers

Because `lpadmin` deletes and modifies files examined by the line printer scheduler, stop the scheduler when you use the `lpadmin` command to remove a printer from lpss.

The following example removes a printer named slap, and moves pending requests to a printer named deke:

Step 1 Ensure that you have superuser capabilities.

Step 2 Deny any further print requests with the `reject` command. Enter:

```
reject -r"decommisioning slap" slap
```

The `-r` option requires a message which enables you to communicate printer status.

Step 3 Disable the printer. Enter:

```
/usr/bin/disable -c slap
```

The `-c` option cancels jobs currently printing on slap.

Step 4 Stop lpss. Enter:

```
/usr/sbin/lpshut
```

Before you stop the line printer scheduler (spooling system), beware of the following:

- All printing stops until you restart the scheduler
- Any print requests currently printing completely reprint when you restart the scheduler. This includes the print requests that were printing page 9,999 of a 10,000 page printout

Step 5 (If you are removing your only printer, omit this step.) Move all print requests in the request directory for the printer to another printer request directory to preserve any pending print requests. Enter:

```
/usr/sbin/lpmove slap deke
```

Step 6 Remove the printer from your lpss. Enter:

```
/usr/sbin/lpadmin -xslap
```

When you use the `lpadmin` command, do not put any spaces between the options and their respective values.

Step 7 (If you have just removed your only printer, omit this step.) Start lpss:

```
/usr/sbin/lpsched
```

Enabling printers

The `enable` command activates the named printer to print requests accepted by the `lp` command. When the printer is enabled—after being disabled—pending print requests begin printing.

You can issue individual commands for each printer or you can combine the printers in one command separated by spaces.

The following example instructs lpss to enable a printer called slap:

Step 1 Ensure that you have superuser capabilities.

Step 2 Enable the printer. Enter:

```
/usr/bin/enable slap
```

Disabling printers

Print requests continue to be accepted for a disabled printer unless you have explicitly rejected print requests. See "Rejecting print requests" on page 111 for instructions.

Disabling a printer should be done to make the printer unavailable for a short time, for example, 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 keep accumulating in the print queue, and they do 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 gets a message stating that the printer is not accepting requests.

When you disable a printer, any print requests waiting to print for that printer remain in the printer's request directory. When the printer is enabled again, the print requests begin printing. Any print requests printing at the time you issue the `disable` command completely reprint 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.

You can issue individual commands for each printer or you can combine the printers in one command separated by spaces. If you combine them, you can also specify different reasons for disabling print requests for different printers.

The following example instructs lpss to disable a printer named slap:

Step 1 Ensure that you have superuser capabilities.

Step 2 Disable the printer. Enter:

```
/usr/bin/disable -r"shutdown for system maintenance" slap
```

The `-r` option requires a message which enables you to communicate printer status.

Managing print requests

This section details managing the lpss after you start the scheduler and add printers. Tasks covered include:

- Controlling print order
- Moving a print request
- Rejecting a print request
- Cancelling a print request
- Changing a print request's priority
- Viewing printer and print request status

Controlling print order

To control the order of printed requests, you can assign priority values to printers and to specific print requests.

Fence priority considerations

The priority is used to select the next spooled file for the targeted printer. If a print request's priority is less than the fence, the minimum priority set for the printer, the request is deferred until the fence is lowered or the priority is raised.

Assigning priorities is not required.

- Priority values must be in the range of 0 to 7. Priority 7 is the highest
- A printer's fence priority determines which print requests get printed. A request's fence priority must be equal to or greater than the printer's fence priority to print. SPP-UX assigns a printer fence priority value of zero (0) when you add a printer to lpss. You can change printer fence priorities with the `lpfence` command
- Each print request has a request priority, which is automatically assigned the destination printer's default request priority. You can change a print request's priority with the `lpalt` command. See "Changing print request priority" on page 112 for more information
- Print request priorities lower than the printer priority do not print. If a print request's priority is lower than its printer's priority, it remains in the request directory —printer queue— for that printer. The request remains there until:
  - Its priority is raised
  - Its printer's priority is lowered
  - The request is canceled
- If multiple print requests wait to print on a specific printer and all have priorities high enough to print:

- lpss prints next the print request with the highest priority
- If more than one print request has the highest priority, all print requests with that priority print in the order they were received by lpss

Changing a printer's fence priority

The following example changes the fence priority for the printer named slap from 0 (the default value) to 7 (the highest value):

Step 1 Ensure that you have superuser capabilities.

Step 2 Stop lpss. Enter:

```
/usr/sbin/lpshut
```

Step 3 Change the fence priority. Enter:

```
/usr/sbin/lpfence slap 7
```

Step 4 Restart the line printer scheduler. Enter:

```
/usr/sbin/lpsched
```

Moving print requests

The `lpmove` command moves print requests from one printer to another. The following example moves all print requests for the printer named slap to the request directory of a printer named deke:

Step 1 Ensure that you have superuser capabilities.

Step 2 Prohibit further requests from entering the request directory. Enter:

```
/usr/sbin/reject slap
```

You can issue individual commands for each printer or you can combine the printers in one command separated by spaces.

Step 3 Disable the printer. Enter:

```
/usr/bin/disable -r"shutdown for system maintenance" slap
```

The `-r` option requires a message which enables you to communicate printer status.

Step 4 Stop lpss. Enter:

```
/usr/sbin/lpshut
```

Step 5 Relocate print requests. Enter:

```
/usr/sbin/lpmove slap deke
```

Step 6 Restart lpss with the `lpsched` command. Enter:

```
/usr/sbin/lpsched
```

Returning print requests to the source printer

The following example reenables the printer named slap:

Step 1 Ensure that you have superuser capabilities.

Step 2 Reenable the printer. Enter:

```
/usr/bin/enable slap
```

You can issue individual commands for each printer or you can combine the printers in one command separated by spaces.

Step 3 Accept print requests. Enter:

```
/usr/sbin/accept slap
```

Rejecting print requests

Reject print requests when a printer is taken off the system for an extended period of time. Do not reject print requests when making the printer unavailable for a short time. For example, it is not necessary to reject print requests for adding paper or changing the toner cartridge. A minor delay due to these short term services is usually acceptable.

To instruct lpss to reject print requests for a printer use the `reject` command.

You can issue individual commands for each printer or combine the printers in one command separated by spaces. If you combine the commands, you can also specify different reasons for rejecting print requests for different printers.

If you do not specify a reason with the `-r` option, the status requests get the following response:

```
Printer is NOT ACCEPTING requests: Reason is
unknown.
```

The following example rejects print requests for the printer named slap:

Step 1 Ensure that you have superuser capabilities.

Step 2 Reject print requests. Enter:

```
/usr/sbin/reject slap
```

Canceling print requests

You can issue individual `cancel` 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.

To list print request identification numbers, use the `lpstat` command.

The `cancel` command has several options that allow you to:

- Cancel all submitted print requests
- Cancel all requests associated with a particular printer

Changing print request priority

The `lpalt` command alters a request made by a previous `lp` command, if it is not currently printing. The primary reasons for changing a print request priority include:

- To move the print request ahead of other requests within the request directory
- To match or exceed the printer's priority, enabling the print request to print

By default the `lp` command assigns your print request a priority equal to the printer's priority setting.

The following example changes the priority for the print request slap-0 to 7:

Step 1 Ensure that you have superuser capabilities.

Step 2 Determine the print request id number and the new priority.

To change the priority of an existing print request you first need the following information:

- The print request's identification number assigned by the system and written to standard output when you submitted the print request
- The new priority value you want to assign (0 -7)

Step 3 Change the priority. Enter:

```
/usr/bin/lpalt slap-0 -p7
```

Viewing printer and print request status

To view the status of printers and print requests, use the `lpstat` command. With no options, `lpstat` displays the status of all requests made by the user. The `-t` option lists the following additional information:

- Status of lpss
- System default printer
- 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

For more information refer to the lpstat(1) man page.

Using accounting

9

SPP-UX allows concurrent sharing of computer resources among users; several users can be logged in, all sharing processors, disk space, and memory. SPP-UX system accounting provides the means to:

- Monitor disk space usage for individual users
- Record connect session data—logins and logouts
- Collect resource utilization data—such as memory usage and execution times—for individual processes
- Charge fees to specific users
- Generate summary files and report used to analyze system performance and bill users for resource consumption

Changes specific to this release

Accounting commands and files moved to conform to new file system layout. See Table 19 for a mapping of previous locations to current locations.

The runacct(1M) command argument `CONNECT` replaces `CONNECT1` and `CONNECT2`. The `CONNECT` entry point is identical to the old `CONNECT1` entry point. See the runacct(1M) man page for more information.

Table 19 Accounting command and file locations

File type	In SPP-UX V4.2	In SPP-UX V5.1 and V5.2
Data	/usr/adm/acct	/var/adm/acct
Commands	/usr/lib/acct	/usr/sbin/acct
Holidays	/usr/lib/acct/holidays	/etc/acct/holidays
Startup	/etc/rc	/etc/rc.config.d/acct

Accounting commands

System accounting commands create, display, remove, merge, and summarize and report data. In addition, the output of one command may become the input to other commands.

For more information check the associated man page for any of the following accounting commands and files:

- acct(1M)
- acctcms(1M)
- acctcom(1M)
- acctcon(1M)
- acctmerg(1M)
- acctprc(1M)
- acctsh(1M)
- diskusg(1M
- fwtmp(1M)
- runacct(1M)
- acct(2)
- acct(4)
- utmp(4)

Setting up and managing licenses

10

Correct operation of your Exemplar S-Class Technical Server requires a license key for each software product. If your users do not acquire a license, they are instructed to contact you for assistance. This chapter:

- Introduces FlexLM
- Summarizes licensing administration
- Points you to WWW resources for obtaining license keys

See the *FlexLM End User Manual* for more information on the licensing program.

Overview

Your software products are licensed using the FlexLM licensing system. Refer to the *Exemplar Distribution Notice* that accompanies your system and the *FlexLM End User Manual* for licensing information specific to your version of the operating system.

Types of licenses

There are two types of licenses for Exemplar Technical Server software products:

- Per CPU—Software is licensed for use on a system with no more than a given number of CPUs
- Per User—Software is licensed for a specific number of simultaneous users. The same UID can run any number of copies, and it only counts as one user. The largest license allows unlimited simultaneous executions. The per user license is applied as follows:

- For SPP-UX—A user is a nonsystem UID with at least one process executing on the system. Nonsystem UIDs are generally UIDs greater than 99
- For layered products—A user is a unique UID running that program. A single user running multiple copies of a program consumes only a single license. This allows operations like a parallel make or multiple CXdb invocations for debugging HP PVM and HP MPI programs

Table 20 lists software products, required license types, and capacities.

Table 20 Software products and licensing policies

Product	Policy	Capacity
SPP-UX	Per User	2, 8, 16, 32, 64, unlimited
NQS	Per CPU	4, 8, 16, 64, 128
Subcomplex manager	Per CPU	4, 8, 16, 64, 128
Fortran	Per User	1, 5, 10, 20, unlimited
C	Per User	1, 5, 10, 20, unlimited
AOP	Per User	1, 5, 10, 20, unlimited
Mlib	Per CPU	4, 8, 16, 64, 128
CXdb	Per User	1, 5, 10, 20, 50, unlimited
CXpa	Per User	1, 5, 10, 20, 50, unlimited
HP PVM and HP MPI	Per User	1, 5, 10, 20, 50, unlimited

How licensing works

When you attempt to use a licensed product, FlexLM receives a request in the form of a key—strings of printable ASCII characters generated by an encryption process. If the key is matched in the FlexLM database, you are allowed to use the product. If the key is not located, you are refused service and a message prompts you to contact your system administrator.

You can obtain keys in any of the following ways:

- Software shipments include keys printed on paper
- Keys are available online through the World Wide Web (WWW)
- TAC provides keys by phone or fax

License administration

The following license administration tools are located in the /usr/local/flexlm directory:

- `lmcksum`—Checksum the license file
- `lmdown`—Graceful shutdown of all license daemons
- `lmgrd`—Flexible license manager daemon
- `lmhostid`—Report the hostid of a system
- `lmremove`—Remove specific licenses and return them to license pool
- `lmreread`—Tell the license daemon to reread the license file
- `lmstat`—Report status on license manager daemons and feature usage
- `lmutil`—Generic FlexLM utility program
- `lmver`—Report the FlexLM version of a library or binary file

Online man pages are available on systems running FlexLM.

Note

You must set the environment variable LM_LICENSE_FILE to /usr/local/flexlm/licenses/convex.dat.

Adding a license key

Perform the following steps to enable a license key for a running system:

Step 1 Add the new license key to /usr/local/flexlm/licenses/convex.dat.

See "Activating a license" for more information. The convex.dat file must be in /usr/local/flexlm/licenses, do not relocate it.

Step 2 Reread the license file. Enter:

```
/usr/local/flexlm/lmreread
```

Step 3 Load the new license key into the kernel. Enter:

```
/etc/gscl
```

Activating a license

To activate the license for a software product, edit the file /usr/local/flexlm/licenses/convex.dat. The convex.dat file must be in /usr/local/flexlm/licenses. Do not relocate it.

Add the license key to this file on a new line, after the SERVER statement and the DAEMON statement as shown in the following sample file:

```
SERVER demospp 12063 765
DAEMON convex_ls /usr/local/flexlm/convex_ls
FEATURE sppux convex_ls 99.990 1-jan-2099 0 EB2E4DFDFB40BC2F8F75 \
        VENDOR_STRING=UIDS=0;PART=710-98765-005 HOSTID=12063 ck=54
FEATURE scm convex_ls 99.990 1-jan-2099 0 9B6EEDFDED37F6C0621 \
        VENDOR_STRING=CPUS=128;PART=710-123450-004 HOSTID=12063 ck=34
FEATURE cc convex_ls 99.990 1-jan-2099 0 8B327A788A35FFED135B \
        VENDOR_STRING=;PART=123-000290-006 HOSTID=12063 DUP_GROUP=U
ck=32
```

Each license key consists of two lines with a continuation symbol or backslash (\) at the end of the first line.

License messages

Unsuccessful attempts to use SPP-UX or layered products produce messages of the form:

`Unable to obtain license for` *product*: *further detail*

where

product

Is the product name.

further detail

Is either a description of the exact problem or detailed error information.

For example:

`Unable to obtain license for spp-ux: Too many unique uids active`

The system that has reached the licensed user limit.

`Unable to obtain license for nqs: Complex size (16) exceeds license (8)`

An NQS license for an 8-CPU complex is running on a 16-CPU complex.

`Unable to obtain license for cxdb: Licensed number of users already reached (-4,132)`

The licensed number of users are already using CXdb.

The numbers in parenthesis are FlexLM error codes.

Obtaining license keys via the WWW

If you subscribe to the Hewlett-Packard Convex web service, you can access keys for your Exemplar Technical Server at the following URL:

http://www.convex.com:90/cust-bin/keys

Enter your username and the password you received upon subscribing.

If you wish to become a subscriber, go to the following URL:

http://www.convex.com:90/subscribe.html

Backing up your system 11

This chapter covers prebackup considerations and instructs you how to:

- Back up files
- List files on a backup tape
- Recover files

Prebackup considerations

Before you begin to back up your data, you need to decide:

- What tape device to use
- What information you wish to backup
- If you want to use a graph file
- What level backup you want: full or incremental
- Whether to allow system access during backups

For more information on `fbackup`, see the fbackup(1M) man page.

Tape devices

Tape device files are contained in either the /dev/rmt or the /dev/tmid directory. Device files located in /dev/rmt have the following Hewlett-Packard DAT tape device file format:

```
0[m | c][n]
```

where

0

Is the numerical address to which the tape drive maps.

`m`

Indicates medium density-1600dpi.

`c`

Indicates data compression—GCR format, 3200BPI.

`n`

Indicates a no-rewind device.

Non-HP tape device file syntax is:

tape_typeND`[c|i] [n | cp | stk]`

where

tape_type

Is the tape device type. See Table 21 for valid prefixes.

Table 21 Valid device file prefixes

Device file prefix	Device type
`rdat` or `dat`	DAT (Digital Audio Tape)
`rtc` or `tc`	3480, 3490, 3490E, and Timberline 9490
`rdlt` or `dlt`	DLT (Digital Linear Tape)
`rctp` or `ctp`	CTP C590 (NTP)
`rrdw` or `rdw`	Redwood, D3
`rmt`	Nine-track magnetic tape
`generic`	Generic tape

N

Is the numerical address to which the tape drive maps.

`c` or `i`

Compresses data—GCR format, 3200BPI.

`n`

Indicates a no-rewind device.

`cp`

Indicates control port.

`stk`

Indicates a changer device.

A tape device located at address 0 which does not rewind your tape after finishing is:

`/dev/tmid/rdat0n`

What to back up?

In order to completely recover from data loss, you need to make copies of all:

- User files
- System files:
  - You modified
  - Added since you originally installed SPP-UX
- Products installed since you originally installed SPP-UX

Graph files

A graph file contains a list of directories and files you want backed up. The text file lists included and excluded files with one entry per line. Each entry must begin with an `i` or `e` followed by the path name of the file you want included or excluded.

For example, to back up all of /usr except for the /usr/lib directory, create a file with the following entries:

```
i /usr
e /usr/lib
```

Backup levels

Once you have identified a list of files to include and exclude from your backup, decide whether you want:

- A full backup—Backs up all of the files on your list
- An incremental backup—Backs up only those files that have changed or been created since the last back up

If you use `fbackup` for incremental backups, you must keep a record of past backups. By default, `fbackup` maintains this data in a text file—/var/adm/fbackupfiles/dates.

You must create the directory—/var/adm/fbackupfiles—prior to the first time you use `fbackup` for incremental backups.

System access during backups

When absolute consistency on a full backup is important, your system should be in single-user mode.

You can do incremental backups while the system is in normal use to improve system up-time.

Backing up files

The following example uses `fbackup` to back up the contents of the /home directory:

Step 1 Log in as root.

Step 2 Ensure that files you want to back up are not being accessed.

The `fbackup` command will not back up files that are active or locked. `fbackup` is designed to allow backups while the system is in use by providing the capability to retry an active file.

Step 3 Load the backup device with a write-enabled tape.

If your backup requires additional tapes, `fbackup` prompts you to load or change tapes.

Step 4 Create the backup using `fbackup`.

For instance, the following command backs up the entire contents of /home to the device file /dev/rmt/0m and writes a list, or index, of all backed up information to /tmp/index:

```
fbackup -f /dev/rmt/0m -i /home -I /tmp/index
```

The index file consists of one line for each file backed up during the session.

Viewing files on a backup tape

If you choose not to create an online index when you back up your system with the `fbackup` command, you can use the `frecover` command's `-I` option to read a backup tape's index.

Step 1 Log in as root.

Step 2 Load the selected write *disabled* tape into the drive.

Remove the write ring or open the write protect tab.

Step 3 Extract the index using the `frecover` command.

For instance, the following command extracts the index from the tape loaded on the device file /dev/rmt/0m and writes it to /tmp/index:

```
frecover -I /tmp/index -f /dev/rmt/0m
```

Step 4 View the index file with your favorite editor or pager.

See the man page for more(1) or less(1) for more information on pagers.

Restoring files

Use the `frecover` command to restore your backup files created with the `fbackup` command. You can restore a whole directory, a

specific file, or an entire backup tape. See the frecover(1M) man page for more information.

If you did not use `fbackup` to create your backup tapes you can not use `frecover` to extract files from your backup tape.

Step 1 Log in as root.

Step 2 Ensure that files you want to restore are not being accessed.

Step 3 Load the backup device with a write *disabled* tape containing the files you want to restore.

Remove the write ring or open the write protect tab.

Step 4 Extract the files using the `frecover` command.

For instance, the following command restores the contents of /users from the tape located on /dev/rmt/0m:

```
frecover -i /users -f /dev/rmt/0m
```

Also see the man pages for dump(1m) and restore(1m) for information on backing up entire filesystems.

Using crashsystem 12

Once you have created a crashdump disk partition with the `diskutil` command, the `crashsystem` 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 in some cases to help determine the cause of the system crash.

When a system crash occurs for a reason you do not understand, contact the Hewlett-Packard Convex Technical Assistance Center (TAC). If the TAC determines that you should send a crashdump file:

- Initiate the crashdump from the test station
- Reset and reboot your system when the crashdump finishes
- Create a crashdump file using `crashutil`
- Make a tape containing the crashdump file
- Mail it to the TAC. See "Technical assistance" on page xvii for information about contacting the TAC

The first three steps are covered in this chapter, as well as information you need to create crashdump partitions.

See "System tunables," chapter on page 143 for crash related tunables.

Changes specific to this release

The `crashutil` command is located in /sbin.

The `crashsystem` command is located on the test station in /spp/bin.

Terms

Terms used in this chapter include:

SCSI (Small Computer System Interface)

An I/O bus protocol using a parallel bus connection between a peripherals. The SCSI protocol is the logical capability of the bus, rather than physical characteristics of peripheral devices.

High-Priority Machine Check (HPMC)

Similar to a trap. Indicates that a process error occurred and that the process cannot continue. In most cases, the system does not require rebooting after an HPMC

Transfer Of Control (TOC)

Is the highest level interrupt on an Exemplar Technical Server. A TOC can be triggered through software or hardware (for instance, pushing the reset button on the chassis.)

interrupt

An occurrence that changes the normal flow of instruction execution. An interrupt originates from hardware, such as an I/O device, or software.

Requirements and restrictions

To use `crashsystem`:

- You must have a designated crashdump partition for each node. To see if a node already has a crashdump partition, enter:

  `/sbin/crashutil -p`

  The system returns a list of nodes and assigned disk partitions. See "crashutil examples" on page 140 for Exemplar S-Class and X-Class Technical Server and SPP-1200/SPP-1600 sample output
- You must set each crashdump partition's crashdump flag with `diskutil`. See "Administering crashdump partitions" on page 133 for instructions
- You may only have one partition on a node marked as a crashdump partition. If you have more than one, unset the extra "C" flag with `diskutil`
- You must have set the crashdump partition size as follows:
  - SPP-1200 and SPP-1600 should have approximately:
    * 50Mb on single-hypernode machines
    * 75Mb on multi-hypernode machines
  - Exemplar S-Class and X-Class Technical Servers should have approximately 128Mb on both single-hypernode and multi-hypernode machines

  The amount of space needed depends on the amount of global shared memory on your system, the amount of memory used at the time of the system crash, and the rate of compression you achieve
- You must locate crashdump partitions in the first 2Gb of the available disk
- You must create a crashdump partition on each hypernode's local disk, in order for that partition to participate in a crashdump
- You must perform a system reboot in order for your system to recognize any changes made to crash partitions
- (Single node only) You must locate crashdump partitions on the root disk
- (Exemplar S-Class and X-Class Technical Servers only) You must install and enable FCode revision 5 firmware. If you do not and your disk's SCSI controller hangs, writes to partitions on that disk are unsuccessful

How crashsystem works

The `crashsystem` utility gains control of the processors on your system by sending a software interrupt. The interrupt class depends on your machine type; see either of the following sections for specifics:

- "SPP-1200 and SPP-1600 machines"
- "Exemplar S-Class and X-Class Technical Servers"

What is a crashdump

A crashdump provides the information necessary to troubleshoot a problem which resulted in a system crash. It contains the following system information:

- CPU register states
- Process tables
- Snapshot of all memory, including:
  - Kernel
  - Server
  - Processes

SPP-1200 and SPP-1600 machines

The crashsystem works by sending a High-Priority Machine Check (HPMC) to gain control of the processors. So, a condition that prevents a processor from receiving an HPMC can prevent crashdump from starting.

For example, if you are unable to read and write to the system's memory from the test station, the crashdump will not work. Also, if a processor has already received an HPMC, it is likely that the crashdump will not work.

Exemplar S-Class and X-Class Technical Servers

The crashdump process starts by sending a Transfer of Control (TOC), from the test station, to gain control of the processors on each node. A packet is generated on the runway bus, much like an HPMC. OBP then:

- Receives the packet
- Saves the CPU register state
- Turns control over to the operating system's TOC handler

Upon establishing control, one CPU (selected by the process) dumps all physical memory in use by the kernel out to disk.

Administering crashdump partitions

When you execute `crashsystem`, each hypernode with a crashdump partition on a directly attached disk writes operating system dump information to the partition. After all hypernodes perform the crashdump, the boot hypernode returns to OBP and you can reset and reboot the system.

Each hypernode should have a crashdump partition. You can use `crashutil` to list existing crashdump partitions. Partitions are created and managed using `diskutil`; see "Creating a crashdump partition" on page 134 for an example.

You must have set the crashdump partition size as follows:

- SPP-1200 and SPP-1600 should have approximately:
  - 50Mb on single-hypernode machines
  - 75Mb on multi-hypernode machines
- Exemplar S-Class and X-Class Technical Servers should have approximately 128Mb on both single-hypernode and multi-hypernode machines

  The amount of space needed depends on the amount of global shared memory on your system, the amount of memory used at the time of the system crash, and the rate of compression you achieve

OBP requires the crashdump partition to be located in the first 2Gb of the disk.

You must reboot the operating system before newly created or changed crashdump partitions are recognized.

This section explains how to use `diskutil` to:

- Create crashdump partitions
- Change crashdump partitions
- Delete crashdump partitions

For more information on `diskutil`, see the "Using diskutil," chapter on page 40.

Creating a crashdump partition

The following procedure creates a crashdump partition, between partitions b and d, on the disk labeled sd0:

Step 1 Log in as root.

Step 2 Start `diskutil`. Enter:

```
/usr/sbin/diskutil
```

Step 3 Show available disks. Enter:

```
DiskUtil: show disks

SD 0:0:2:0 mapped to sd7
SD 0:0:3:0 mapped to sd0
SD 0:0:4:0 mapped to sd19
SD 0:0:5:0 mapped to sd16
SD 0:0:9:0 mapped to sd17
SD 0:0:10:0 mapped to sd1
```

Step 4 Select the disk.

```
DiskUtil: select disk sd0
```

In this example, you selected the disk labeled `sd0`.

Step 5 Show the partitions. Enter:

```
DiskUtil: show partitions
Logical disk name: sd0
partition table: (space available for file systems = 2098759)
part    offset      size    | partition description          | flags
-------------------------------------------------------------------
 a:          0K   819200K  |Root and /usr filesystem        | *
 b:     819200K  1048576K  |Default Pager Partition         | *D
 d:    1908736K   190023K  |/tmp filesystem                 | *
```

In the previous output, disk sd0 has free space between partitions b and d.

Step 6 Use the `make partition` command to create a crashdump partition. For example:

```
DiskUtil: make partition c size size after b description "crash"
```

where *size* is the appropriate size in bytes for your machine type. `diskutil` does not let you make the partition larger that available space, in other words, you cannot overrun the next partition.

Step 7 Set the crashdump flag for this partition with the `set partition` command. For example:

```
DiskUtil: set partition c flag crashdump description "crashdump"
```

Step 8 Reboot the machine after exiting `diskutil`.

Changing or deleting a crashdump partition

Instead of creating a new crashdump partition, you can change an existing partition to a crashdump partition:

Step 1 First make sure that all useful data has been removed from the partition.

Step 2 Ensure that the selected partition is not mounted.

Step 3 Use the `set partition` command to identify this partition as a crashdump partition. Enter:

```
DiskUtil: set partition c flag crashdump
```

To stop using a partition as a crashdump partition, use the `unset partition` command. Enter:

```
DiskUtil: unset partition c flag crashdump
```

Caution

Do not create more than one crashdump partition on a single node; this causes the crashsystem to fail.

In order for your system to recognize any changes made to crash partitions, you must perform a system reboot.

For more information see "Using diskutil" on page 40.

Starting and monitoring a crashdump

Initiating and monitoring a crashdump differs depending on your machine type. Exemplar Technical Servers use the `do_reset` and `sppconsole` commands and SPP-1200/SPP-1600 systems use the `crashsystem` and `sn_cnsl` commands.

On your Exemplar Technical Server

To initiate and monitor a crashdump for your Exemplar Technical Server, use the `do_reset` and `sppconsole` commands as shown in the following steps.

Step 1 From the test station, start the crashdump with `do_reset`. Enter:

`/spp/bin/do_reset` `[`**`all`** | *node_id*`]` **`4`**

where

`all`

Resets all nodes in contact with the test station.

node_id

Is a specific node number. Requires the level argument.

`4`

Is the level argument. Level `4` sends a Transfer Of Control (TOC) to initiate a crashdump. Requires the *node_id* argument.

Depending on the size of your system, the crashdump process may take five to ten minutes to complete.

Step 2 If you have a test station, monitor the crashdump with `sppconsole`.

Syntax is:

`/spp/scripts/sppconsole` *hostname*

where *hostname* is the machine name. The command's default mode watches, or spys, on the console. Table 22 lists commands you enter while you run `sppconsole`.

Table 22 Interactive sppconsole commands

To...	Enter...
Take control of the console (Enables you to write to the console.)	CTRL-Ecf
Change an active console window to a spy window.	CTRL-cs
Exit sppconsole.	CTRL-Ec. (The period is part of the command.)

For remote nodes, log in to the node's test station and run sppconsole. If the node does not have a test station, you can not see the crashdump output.

Step 3 Reset and reboot your system.

Upon completion, the boot hypernode reenters OBP and the ok prompt appears on the system console. You can then reset and reboot your Exemplar Technical Server.

Step 4 Use crashutil to convert raw crashdump data to a formatted file.

Execute crashutil and specify a unique output file name for each existing crashdump partition. For more information see "Using crashutil" on page 139 and the crashsystem(1M) man page.

On your SPP-1200 or SPP-1600

To initiate and monitor a crashdump for your SPP-1200 or SPP-1600 machine, use the crashsystem and sn_cnsl commands as shown in the following steps:

Step 1 Start the crashdump with crashsystem. Enter:

```
/spp/bin/crashsystem all
```

where all resets all nodes in contact with the test station.

Depending on the size of your system, the crashdump process may take five to ten minutes to complete.

Step 2 Monitor the crashdump with sn_cnsl.

The sn_cnsl command allows you to either remotely view the SPP-UX console window or to remotely assume control. You must be logged in to the test station, either locally or remotely, as root in order to use the sn_cnsl command. Command syntax is:

`/spp/bin/sn_cnsl [-fFsS]` *console_id*

`-f`

Force control of the SPP-UX system console to the host from which the command was entered.

`-F`

Force control of the SPP-UX system console to the host from which the command was entered; also print the console messages stored in the message buffer.

`-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.

`-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; also print the console messages stored in the message buffer.

console_id

Identifies the console you wish to view; the default value is 1. If your system's root node is not node 0, *console_id* is the root node number + 1. Output for node 0 goes to the main console, output from other hypernodes goes to virtual consoles. Virtual console numbering starts at 2091 for hypernode 1 and increments by one for each hypernode.

Table 23 details commands you enter while you run `sn_cnsl`.

Table 23 Interactive `sn_cnsl` commands

To...	Enter...
Take control of the console (Enables you to write to the console.)	**CTRL-Ecf**
Change an active console window to a spy window.	**CTRL-Ecs**
Exit `sn_cnsl`.	**CTRL-Ec.** (The period is part of the command.)

Step 3 Reset and reboot your system.

Upon completion, the boot hypernode reenters OBP and the `ok` prompt appears on the system console. You can then reset and reboot your system.

Step 4 Use `crashutil` to convert raw crashdump data to a formatted file.

Execute `crashutil` and specify a unique output file name for each existing crashdump partition. For more information see "Using crashutil" on page 139 and the crashsystem(1M) man page.

Using `crashutil`

If the TAC determines that you should send a crashdump, use the `crashutil` command to create a crashdump file.

Each crashdump overwrites all crashdump data from previous crashdumps, if any exist. Run `crashutil` on your system to retrieve crashdump data from the crashdump partition before initiating another crashdump.

Default file compression

Due to the large amount of space they consume (20-100Mb), crash files are compressed by default. Output is compressed by `gzip`, which appends `.gz` to the filename.

If `gzip` is not available, you can use the following `crashutil` options:

`-n`

Generates an uncompressed crashdump file.

`-g`

Specifies an alternate compression program.

`crashutil` syntax

The `crashutil` command has the following format:

`crashutil -d -n -p [-i` *crashdump_partition* | *crashdump_file*] `[-o` *outfile*] `[-g` *compression*]

`-d`

Display the contents of the crashdump partition or file designated with the `-i` option.

`-n`

Do not compress the output file. If this option is not specified, the crashdump file is compressed using `gzip` and a `.gz` suffix is appended to the crashdump file.

`-p`

Display all possible crashdump partitions on the machine, along with their node numbers and time stamps. You can have only one partition on a node marked as the crash partition.

`-i` [*crashdump_parition* | *crashdump_file*]

Is the input crash partition or file.

`-o` *crashdump_file*

Is the output crashdump file. (Requires the `-i` *crashdump_partition* option.)

`-g` *compression*

Is the path to the alternate program to compress the crashdump output file. The program should read uncompressed data on standard input and write compressed data to standard output. When this option is specified `.gz` is not appended to the output filename.

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

`crashutil` examples

The output you receive from the `crashutil` command differs between Exemplar S-Class and X-Class Technical Servers and SPP-1200/SPP-1600 machines.

Exemplar Technical Server

To display all crashdump partitions on an Exemplar Technical Server, enter:

```
crashutil -p
```

```
Current crashdump partitions:  last date written
        node 0 : sd2e   Fri Mar 14 15:05:47 1997
        node 1 : sd21c  Fri Oct  4 23:01:36 1996
```

To copy a crash file from a crashdump partition to a file on your Exemplar Technical Server, enter:

```
crashutil -i  sd2e -o crashfile.0
```

SPP-1200 and SPP-1600

To display all crashdump partitions on an SPP-1200 or SPP-1600 machine, enter:

```
crashutil -p
```

```
Current crashdump partitions:  last date written
        node 0 : sd3b   Mon Apr 21 14:20:33 1997
```

To copy a crash file from a crashdump partition to a file on your SPP-1200 or SPP-1600 machine, enter:

```
crashutil -i sd3b -o crashfile.0
```

System tunables

A

The /stand/spp3/tunables file allows you to adjust system performance with tunable parameters. The parameters from the this file are read and set for the system each time SPP-UX boots. If you change the values for parameters in /stand/spp3/tunables, the new values take effect the next time the system boots.

Changes specific to this release

The tunables file is located on your Exemplar S-Class Technical Server in /stand/spp3/tunables.

The information formerly found in /etc/newconfig/tunables is now located in /usr/newconfig/stand/tunables.

The tunables file

The tunables file contains parameters for both the SPP-UX microkernel and SPP-UX server.

When you install a new version of SPP-UX, your /stand/spp3/tunables file is not automatically updated to add new tunables. Instead, a new tunables file is installed in the /usr/newconfig/stand directory. After installing a new version of SPP-UX, check the /usr/newconfig/stand/tunables file against your /stand/tunables file to see if new tunables have been added or if the range of values for tunables has changed.

The following sample illustrates the format of the /stand/spp3/tunables file:

```
#       $CHeader: tunables.cnx_mpp1 1.20 1996/01/08 11:51:53 $
#
#       Copyright 1993-1996, CONVEX Computer Corporation.
#       This document is copyrighted.  All rights are reserved.  This
#       document may not, in whole or part, be copied, duplicated,
#       reproduced, translated, electronically stored or reduced to
#       machine readable form without prior written consent from CONVEX
```

```
#       Computer Corporation.

################################################################################
# There are many tunables that are not in this file. They take their
# default values when the system boots. For more information about
# tunables, please see the online tunables manual page.
#
# Command:    man tunables
################################################################################
#
# Knowledge of the internal operations of SPP-UX is recommended before
# altering these values.  The following formats for values are accepted -
#       s       string
#       n       decimal numeric
#       0xn     hex numeric
#       nK      n * 1024
#       nM      n * 1024 * 1024
#
# Where ranges are present in the key, such as "1[0..15]", this generally
# refers to the values on each of the possible nodes in the system.  With
# this format all entries in the array range are set to the single value
# specified.
#
#
#       Microkernel tunables
#
# Event Logger internal buffer, for holding events before syslog becomes active
Event Logger,buffer size:desc=Event Logger Buffer Size:1=64k:default=64k:
Event Logger,event logger threshold:desc=Event Logger Condition Code Threshold:1=0:default
=0:
Event Logger,event logger boot threshold:desc=Event Logger Condition Code Boot Threshold:1
=2:default=2:
# Control the LCD processor heartbeat update (0:Off, 1:On)
LCD Heart Beat,control:desc=Control for LCD update:1=1:default=1:
# Spin lock consistency checking (0:Off, 1:On)
Spin Locks,lock checks:desc=Enable spin lock consistency checks:1=1:default=1:
#
#       Server tunables
#
# Percent of memory dedicated to the filesystem buffer cache
Fileserver,buffer_cache_percent:1[0..15]=10:
.
.
.
```

Tunables

Each of the following tables describes the tunables controlling a particular component of SPP-UX. The value is written in the tunables file with the following format:

facility_name, tunable:desc:init:default

where

facility_name

Is the system to which the tunable applies. Values include: `Event Logger`, `Crashdump`, `Fileserver`, `Server`, `Emulator`, `Routearpserv`, `Spin Locks`, `Stripe Disk`, `Virtual Memory`, `LCD Heart Beat`, and `Landmarc`.

tunable

Is the tunable parameter name.

desc

Is a brief description.

init

Is the setting for your system.

default

(Optional) Is the default value.

Precede the tunables listed in Table 24 with the facility name, `Crashdump`.

Table 24 Crashdump tunables

Tunable	Range	Default	Description
Crashdump, crash_debug	0 - 1	0	Wait for kgdb connection instead of prompting for crashdump on panic
Crashdump, panic_graceful	0 - 1	1	Determines whether messages are printed when a microkernel panic occurs
Crashdump, panic_query	0 - 1	0	Determines whether operator is prompted for crashdump tunables at the time of panic
Crashdump, panic_reboot	0 - 1	0	Reboot automatically after dump completes. Requires OBP `autoreboot` to be configured

Precede the tunables listed in Table 25 with the facility name, `Event Logger`.

Table 25 Event logger tunables

Tunable	Range	Default	Description
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
boot threshold	0 - (7 << 29)	0	Minimum severity level for threshold messages logged at boot time
threshold	0 - (7 << 29)	0	Minimum severity level for messages logged at any time after the boot process

Precede the tunables listed in Table 26 with the facility name, `Fileserver`.

Table 26 File server tunables

Tunable	Range	Default	Description
buffer_cache_percent	1 - 80	10	Buffer cache for each node specified as a percent of physical memory
disk_wdb_size	0 - (128*1024)	1024	Bytes allocated for wired device buffers for hard disk drives, a multiple of 1024
tape_wdb_size	0 - (16*1024*1024)	128*1024	Bytes allocated for wired device buffers for tape drives, a multiple of 1024

Precede the tunables listed in Table 27 with the facility name, `Server`.

Table 27 Process server tunables

Tunable	Range	Default	Description
distribute_panic	0 | 1	1	Sets (1) or clears (0) a flag that determines whether a panic in the node's server is distributed to all other nodes' servers, or whether it causes only the local node's server to shut down
acctresume	0 - 100	4	Percentage of file system space that must be free to reactivate process accounting after it is suspended due to insufficient free space

Table 27 Process server tunables **(continued)**

Tunable	Range	Default	Description
acctsuspend	0 - 100	2	Percentage of file system space that must be free in order to allow process accounting
dfldsiz	512K - 0xC0000000	64*1024*1024	Default number of bytes in a process's data segment, a multiple of page size
dst	0 | 1	1	Sets (1) or clears (0) a flag that specifies whether daylight savings time will be used
incksum	1 | 2 | 3	2	Internet Protocol (IP) checksum method. Set to 2 under normal circumstances. If `netstat -s` shows an abnormally high number of IP, TCP, or UDP checksum errors, contact the TAC for assistance in changing this value
maxdsiz	512K - 3G	512M	Maximum number of bytes in a process's data segment, a multiple of page size
maxfiles	1 - 256	256	Maximum number of files a process can have open at once. This limit can be changed by using `setrlimit`
maxssiz	512K - 3G	512M	Maximum number of bytes in a process's stack, a multiple of page size
maxuprc	8 - 1024	256	Maximum number of processes a user can have at once
maxusers	8 - 1024	256	Maximum number of users; used to compute system parameters
msgmax	0 - 64K	8K	Maximum number of queued messages a user can have
msgmnb	0 - 64K	16K	Maximum number of bytes allowed for all queued messages
msgmni	1 - int_max	50	Number of message queue identifiers
msgtql	1 - int_max	40	Number of message queue headers
ncallout	26 - int_max	2084	Number of timeouts that can be pending simultaneously
nfile	32 - int_max	266	Maximum number of open files
nflocks	50 - 400	200	Maximum number of file locks

Table 27 Process server tunables **(continued)**

Tunable	Range	Default	Description
nmount	1 - int_max	20	Maximum number of mounted file systems
nproc	10 - int_max	2068	Maximum number of processes
npty	16 - 27900	60	Maximum number of pseudo-terminals
printtunables	0 | 1	1	If set, print all tunable values to log at boot
semaem	0 - int_max	16K	Maximum value by which a semaphore can be adjusted due to the death of a process
semmni	2 - int_max	64	Number of semaphore identifiers
semmns	2 - int_max	128	Number of semaphores
semmnu	1 - int_max	2068	Maximum number of processes that can have semaphore undo requests on a semaphore
semume	1 - int_max	10	Maximum number of semaphores on which a process can have a pending semaphore undo request
semvmx	1 - 65535	32767	Maximum semaphore value
shmmax	2K - 3G	0x4000000	Maximum number of bytes in a shared memory segment
shmmni	3 - 1024	200	Maximum number of shared memory segments
shmseg	3 - 1024	120	Maximum number of shared memory segments that can be attached to a process at once
timezone	0 - 1500	300	Minutes your timezone is "west" of GMT
unique_core_names	0 | 1	0	If set, each core file name generated by kernel is unique: `core.`*pid.time*

Precede the tunables listed in Table 28 with the facility name, `Emulator`.

Table 28 Emulator tunables

Tunable	Range	Default	Description
print_emul_tunables	0 | 1	0	If set, prints emulator specific tunables to boot log
adv_WS_option	0 | 1	1	If set, enables advertisement of the TCP windowsize
adv_TS_option	0 | 1	0	If set, enables timestamp advertising
disable_loopback_csums	0 | 1	0	Disables checksums on packets sent over the loopback interface
do_udpcksum	0 | 1	0	Enables checksums of UDP packet headers
tcp_loopback_mtu	1000 - 64512	16384	Maximum transmit unit for TCP over the loopback interface
raw_default_socketspace	4096 - 262144	32768	Amount of buffer space set aside for raw socket data
sendmsg_access_rights	0 | 1	1	Enables (1) or disables (0) the sendmsg system call to pass access rights
str_n_event	0 - 512	25	Number of system bufcall structures to allocate
str_n_tevent	0 - 512	128	Number of system timeout structures to allocate
str_n_mblk	0 - 8192	600	Initial number of m_blks allocated
str_lo_pct	0 - 100	60	Percent of mblk's to be low priority
str_med_pct	0 - 100	80	Percent of mblk's to be medium priority
str_msg_sz	16K - 10M	2M	Maximum message size. Default is HIPPI maximum socket size
str_ctl_sz	16K - 64K	16K	Maximum ioctl message size

Precede the tunables listed in Table 29 with the facility name, `Routearpserv`.

Table 29 Routearpserv tunables

Tunable	Range	Default	Description
print_ras_tunables	0 | 1	0	If set, routearpserver-specific tunables are logged
subnetsarelocal	0 | 1	1	If set, an address is local even if it belongs to another subnet.
do_rarp	0 | 1	0	If set, the Exemplar Technical Server acts as a reverse-ARP server
do_ipf	0 | 1	0	If set, the Exemplar Technical Server acts as an IP gateway, if it has multiple interfaces
do_ipsendredirects	0 | 1	1	If set, the Exemplar Technical Server sends ICMP IP route redirect packets to improve route performance

Precede the tunables listed in Table 30 with the facility name, shown in the first column of the table.

Table 30 Miscellaneous subsystem tunables

Facility	Tunable	Range	Default	Description
Spin Locks	lock checks	0 | 1	1	If set, simple locks are in debug mode
Stripe Disk	max stripes	0 - int_max	24	Maximum number of stripes per hypernode
Virtual Memory	max_commit_percent	1 - 100	90	(Change only with assistance of TAC.) Affects amount of virtual memory committed to jobs
Virtual Memory	gsm_0_contrib	0 | 1	0	When set, allows a node to make no contribution to global shared memory of a subcomplex

Table 30 Miscellaneous subsystem tunables **(continued)**

Facility	Tunable	Range	Default	Description
LCD Heart Beat	control	0 | 1	1	Sets (1) or clears (0) a flag that determines whether the control LCD displays a heartbeat and the state of each CPU, updated every 1/4 second
Landmarc	bte_pages	64 - mem_pages/ 8	.05* mem_pages +3620	Pages to allocate for Landmarc Block Table Entries

The `cnx_get_tunable` system call

The `cnx_get_tunable` system call finds the *default* value of a system tunable stored in the /stand/spp3/tunables file. It does not return the current value of a tunable if you have modified the default value.

SPP-UX directories

B

Table 31 lists current SPP-UX directories and maps them to equivalent directories from previous SPP-UX releases.

For a thorough description of the contents of each directory, see the *HP-UX 10.0 File System Layout Whitepaper* located, on your system, in /usr/share/doc/filesys.txt.

Table 31 SPP-UX directory map

Current	Previous	Contents
/dev	/dev	Device files for local devices
/etc	/etc	Machine-specific configuration and administration databases; no executables invoked by users
/etc/opt/*	N/A	Application-specific configuration files
/export	N/A	Default root of exported file systems
/export/private_roots	N/A	Host-specific files
/export/shared_roots	N/A	Shared operating system and applications
/home	/users	Default user directories
/home/*	/users/*	User home directories
/lost+found	/lost+found	Storage directory for fsck
/mnt	/mnt	Mounting point for local file systems
/net	/net	Mounting point for remote file systems
/opt	N/A	Root for optional applications
/opt/*	/usr/*	Application executables, libraries, and support files

Table 31 SPP-UX directory map **(continued)**

Current	Previous	Contents
/sbin	N/A	Essential system commands for booting the system and mounting file systems
/sbin/init.d	N/A	Startup and shutdown scripts
/sbin/rc#.d	N/A	Startup and shutdown link files for script sequencing
/stand	N/A	Standalone machine dependent binaries and kernel configuration
/tmp	/tmp	System generated temporary files
/usr	/usr	Mount point for sharable user commands, libraries, and documentation
/usr/bin	/bin and /usr/bin	Operating system user commands
/usr/ccs	N/A	Unbundled development package
/usr/ccs/bin	N/A	Development binaries
/usr/ccs/lib	N/A	Development libraries
/usr/conf	/etc/conf	Kernel configuration
/usr/contrib	/usr/contrib	Contributed software
/usr/include	/usr/include	Header files
/usr/lbin	N/A	Backends to other commands
/usr/lib	/lib and /usr/lib	Object code and object code libraries
/usr/local	/usr/local	User-contributed software
/usr/newconfig	/etc/newconfig	Default operating system configuration data files
/usr/old	N/A	Obsolete files
/usr/sbin	N/A	System administration commands
/usr/share	N/A	Architecture-independent sharable files
/usr/share/dict	/usr/lib/spell	Dictionaries for spell and ispell
/usr/share/lib	N/A	Miscellaneous
/usr/share/man	/usr/man	Operating system manpages
/var	N/A	Holds files created at runtime (log files and temporary files)
/var/adm	/usr/adm	Common administrative files and log files

Table 31 SPP-UX directory map **(continued)**

Current	Previous	Contents
/var/adm/crash	/tmp/syscore	Kernel crashdumps
/var/adm/cron	/usr/lib/cron	Cron queueing
/var/adm/sw	N/A	SD directory
/var/adm/sw/patch	N/A	Software patch storage directory
/var/adm/syslog	N/A	Files generated by syslog
/var/mail	/usr/mail	Incoming mail
/var/news	/usr/news	News
/var/opt/*	N/A	Application specific temporary or data files
/var/preserve	/usr/preserve	Preserved editor files
/var/run	N/A	PID files
/var/spool	/usr/spool	Spooled files
/var/spool/	/usr/spool/cron	Crontabs and at jobs
/var/spool/locks	/usr/spool/locks	UUCP Lock files
/var/spool/lp	/usr/spool/up	Printer spooling
/var/spool/mqueue	/usr/spool/mqueue	Outgoing mail
/var/spool/sw	N/A	Default location for SD depot
/var/tmp	/usr/tmp	Application-generated temporary files
/var/uucp	/usr/spool/uucp	UUCP administration
/var/spool/uucppublic	/usr/spool/uucppublic	Incoming UUCP files
/var/tmp	/usr/tmp	Application-generated temporary files
/var/uucp	/usr/spool/uucp	UUCP administration

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

S

T

U

V

W

B5655-90023