

HP 9000
Computers

Solving HP-UX Problems

Solving HP-UX Problems



**HP Part No. B2355-90030
Printed in USA August 1992**

**Third Edition
E0892**

Legal Notices

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

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

Warranty. A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

©copyright 1983-92 Hewlett-Packard Company

This document contains information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Restricted Rights Legend. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY
3000 Hanover Street
Palo Alto, California 94304 U.S.A.

Use of this manual and flexible disk(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

©copyright 1980, 1984, 1986 AT&T Technologies, Inc.

UNIX is a registered trademark of Unix System Laboratories Inc. in the USA and other countries.

©copyright 1979, 1980, 1983, 1985-90 Regents of the University of California
This software is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California.

©copyright 1979 Regents of the University of Colorado, A Body Corporate.
This document has been reproduced and modified with the permission of the regents of the University of Colorado, a body corporate.

©copyright 1986, 1987, 1988 Sun Microsystems, Inc.

©copyright 1986 Digital Equipment Corporation.

©copyright 1985-86, 1988 Massachusetts Institute of Technology.

X Window System is a trademark of the Massachusetts Institute of Technology.

MS-DOS and Microsoft are U.S. registered trademarks of Microsoft Corporation.

OSF/Motif is a trademark of the Open Software Foundation, Inc. in the U.S. and other countries. Certification for conformance with OSF/Motif user environment pending.

All rights reserved.

Printing History

The manual printing date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

- Edition 1:September 1989
- Edition 2:January 1991
- Edition 3:August 1992

Preface

The purpose of this manual is to provide you with direction to help you solve many of the problems you may encounter while using and administering HP-UX. It does not provide an exhaustive list of problem/cause descriptions but instead provides methods for locating and solving problems. When you encounter a problem when using HP-UX, use this manual first to guide you toward finding the cause of the problem.

- Chapter 1** *Solving Your Problems* discusses a number of general troubleshooting techniques. Subsequent chapters show how to apply these techniques to specific areas within HP-UX and provide references to further information.
- Chapter 2** *Line Printer Spooling System Problems* assists you in finding and correcting problems with the line printer spooling system.
- Chapter 3** *UUCP Problems* presents information on troubleshooting UUCP problems.
- Chapter 4** *Diskless Cluster Problems* covers problems you may encounter when operating a diskless cluster environment.
- Chapter 5** *System Boot-up* helps you determine why your system won't boot.
- Chapter 6** *File System Problems* discusses how to find and correct errors in the file system structure.
- Chapter 7** *Disk Space Problems* provides a number of possible methods for handling disk space shortages.
- Chapter 8** *Logical Volume Manager (LVM) Problems* discusses problems related to working with volume group activation, booting your system from a LVM disk, and other related LVM problems.
- Chapter 9** *Unresponsive Terminals* describes how to determine the cause of an unresponsive terminal problem.
- Chapter 10** *System Panics* helps you recover from system panics (crashes).



Contents

1. Solving Your Problems	
Recording the Events	1-2
Why keep records?	1-2
What information you should record	1-2
Problem Log Entry:	1-3
Tools to help you record the information	1-4
Defining the Problem	1-4
Isolating the Problem	1-5
Determining the Cause	1-7
Preparing for Problems in Advance	1-8
2. Line Printer Spooling Problems	
An overview of the Line Printer Spooling System	2-1
Using the LP Spooler with HP-UX Clusters	2-3
The Problem Areas	2-4
Spooler Scheduler	2-4
Restarting printouts following a paper jam or paper out	2-5
Restarting printing from the beginning of the job	2-5
Restarting printing from the point where it stopped	2-6
Canceling a runaway printout	2-6
Moving print requests from one printer to another	2-7
Example (Using <code>lpmove</code>)	2-8
What to do when the printer won't print	2-9
Defining the problem:	2-9
Isolating the problem:	2-9
Determining the cause:	2-10
For problems associated with only one printer	2-10
For problems of a global nature	2-12
Where to go for more information	2-12

3. UUCP Problems

An Overview of UUCP	3-1
The Problem Areas	3-1
An example of a UUCP configuration	3-4
SYSTEM #1 System Name: Gold (System Model: HP9000/840)	3-5
Device Files Used:	3-5
Modem Phone Number:	3-5
Entries in /etc/passwd	3-5
Entry in /etc/inittab	3-5
Entries in /usr/lib/uucp/Systems	3-5
Entries in /usr/lib/uucp/Devices	3-6
SYSTEM #2 System Name: Silver (System Model: HP9000/350)	3-6
Device Files Used:	3-6
Modem Phone Number:	3-6
Entries in /etc/passwd	3-6
Entry in /etc/inittab	3-6
Entries in /usr/lib/uucp/Systems	3-6
Entries in /usr/lib/uucp/Devices	3-7
SYSTEM #3 System Name: Bronze (System Model: HP9000/850)	3-7
Device Files Used:	3-7
Entries in /etc/passwd	3-7
Entry in /etc/inittab	3-7
Entries in /usr/lib/uucp/Systems	3-7
Entries in /usr/lib/uucp/Devices	3-8
Defining the problem:	3-8
Isolating the problem	3-8
Step 1: Test the hardware	3-8
Modem Connections	3-8
Direct Connections	3-11
Step 2: Use cu to test /usr/lib/uucp/Devices	3-12
Modem Connections	3-12
Direct Connections	3-14
Step 3: Use cu to test /usr/lib/uucp/Systems	3-15
Step 4: Use uucp to transfer a file	3-17
Obtaining UUCP status information	3-19

Where to go for more information	3-22
4. HP-UX Cluster Problems	
An Overview of HP-UX Clusters	4-1
Problem Areas	4-4
System Boot-up Problems	4-4
Troubleshooting problems with clients that won't boot . . .	4-4
Local Disk Boot-up Problem	4-6
System Panics	4-6
LAN Problems	4-7
CDF Mix-ups	4-7
Configuration/Clustering Problems	4-10
Problems with <code>init</code>	4-10
"Failed kernel selftest" errors	4-11
HP-UX Cluster Log Files	4-12
Allowing Users Shutdown Capabilities	4-12
Where to go for more information	4-14
5. System Boot-up Problems	
What Happens During the System Boot-up Process	5-1
Quick Reference Table (Start Here)	5-2
Phase 1: Boot ROM Initializes Hardware	5-8
What Can Go Wrong During Phase 1	5-8
Symptoms (Phase 1 Problems)	5-8
Phase 2: Boot ROM Loads the First Level of Software	5-11
What Can Go Wrong During Phase 2	5-11
Symptoms (Phase 2 Problems)	5-12
Series 300 and Series 400 Computers	5-12
Series 700 Computers	5-15
Series 800 Computers	5-18
Phase 3: HP-UX is Loaded and Launched	5-20
Symptoms (Phase 3 Problems - Can't Access Kernel)	5-20
Series 300 and Series 400 Computers	5-20
Series 700 and Series 800 Computers	5-20
Symptoms (Phase 3 Problems - Finds Wrong Kernel)	5-23
Series 300/400 Computers	5-23
Series 700 Computers	5-23

Booting from an alternate Kernel on a Series 700 Computer	5-24
Series 800 Computers	5-24
Phase 4: HP-UX Begins Running	5-26
What Can Go Wrong During Phase 4	5-26
Symptoms (Phase 4 Problems)	5-27
This Doesn't Look Like My System (Strange Behavior After Boot)	5-30
System Boots From Local Disk Instead of the Cluster Server	5-31
Series 300/400 Computers	5-31
Series 700 Computers	5-32
Creating/Recreating a Boot Area on a Disk or Logical Volume	5-33
Where to go for more information	5-36

6. File System Problems

File System Overview	6-1
File System Structure	6-1
File System Buffer Caching	6-3
The Problem Areas	6-4
Defining the problem	6-4
Identifying symptoms of a corrupt file system	6-4
Locating and correcting file system corruption	6-5
Example:	6-6
Prerequisites (Record appropriate information)	6-6
Step 1: Terminate processes with files open on the suspect file system	6-8
Step 2: Unmount the suspect file system	6-10
Step 3: Run fsck using preening mode (option -p or -P)	6-11
Step 4a: Check for other causes	6-12
Step 4b: Restore any necessary files	6-12
Step 5: Run fsck interactively	6-13
You can save time by using fsck 's "-y" option	6-14
Step 6: Examine files in the lost+found directory	6-15
Problems with mounting and unmounting file systems	6-16
Mounting Problems:	6-16
Unmounting Problems	6-17
Where to go for more information	6-17

7. Disk Space Problems	
An overview of disk sectioning	7-1
Disk organization on Series 300/400/700 computers	7-1
Disk organization on Series 800 computers	7-1
Traditional Disk Sections	7-2
Logical Volumes	7-3
Locating the space shortage	7-5
Accessing the superuser's extra space	7-6
Resolving disk space problems	7-6
Removing files	7-6
Moving files	7-7
Archiving files	7-7
Creating symbolic links	7-7
Shortening files that grow without bound	7-8
Using Logical Volumes	7-9
Adding new disks	7-9
Preparing for and avoiding disk space problems	7-9
Where to go for more information	7-10
8. Logical Volume Manager (LVM) Problems	
How LVM is Implemented (The LVM Mechanism)	8-2
Traditional Disk Sections vs. Logical Volumes	8-2
Logical Volume Implementation	8-3
Quorum	8-3
LVM Problems, Prevention and Preparation	8-4
LVM Problem Prevention	8-4
Preparing for LVM Problems	8-7
Backing up and Restoring LVM Configuration Information	8-7
LVM Configuration Data (Backing Up)	8-7
LVM Configuration Data (Restoring)	8-8
Creating a "Picture" of Your LVM System Configuration	8-9
The Problem Areas	8-11
Can't Boot From a Logical Volume	8-11
Problems with Missing Physical Volumes During Boot	8-12
Problems with Corrupted LVM/Boot Area on Disk	8-15
Booting an LVM System in Maintenance Mode	8-16
LVM Kernel Trying to Boot from Non-LVM Disk	8-19
Cannot Activate a Volume Group	8-19

I Enlarged My Logical Volume: Why Don't I Have More Space?	8-21
File systems	8-22
Swap Space	8-23
I Reduced the Size of a Logical Volume and Now My System Crashes!	8-23
File Systems	8-23
Swap Space	8-24
What to do if Your System Crashes	8-25
When it is Safe to Reduce the Size of a Logical Volume	8-26
Program Hangs: No Response or Program Output	8-26
How to Create the Picture of Your LVM System Configuration	8-27
Part 1: Your System's Hardware	8-28
Part 2: The Volume Groups on Your System	8-33
Part 3: The Logical Volumes on Your System	8-37
Part 4: The Mirrors on Your System	8-40
Part 5: Putting it all Together (Presenting the Picture)	8-45
Recovering an lvmtab File - Using vgscan	8-50
Reconfiguring Disks in a Volume Group (Adjusting /etc/lvmtab)	8-52
Moving a Volume Group's Disks	8-52
Importing and Exporting Volume Groups	8-53
Where to go for more information	8-54

9. Problems with Terminals

Unresponsive Terminals	9-1
Step 1: Check the status of the system	9-2
Step 2: Check to see if an editor is running on the terminal	9-3
Step 3: Enter ctrl-q at the terminal keyboard	9-4
Step 4: Reset the terminal	9-4
Step 5: Check the terminal configuration	9-5
Step 6: Check the physical connection	9-5
Step 7: Kill processes associated with the problem terminal	9-6
Step 8: Attempt to re-login to the previously hung port	9-7
Step 9: Use cat to send an ASCII file to the hung terminal's device file	9-7
Step 10: Check the parameters of the device file for the problem terminal.	9-8

Step 11: Other things to check	9-8
Make sure your <code>inittab</code> entries are active (<code>telinit -q</code>)	9-8
Check for functioning hardware	9-8
Other Terminal Problems	9-9
What to check for when garbage is mixed with valid data	9-10
What to check for when everything printed is garbage	9-10
Where to go for more information	9-11
10. System Panics	
System Panics: What They Are And Why They Happen	10-1
What to do when your system panics	10-2
Step 1: Record the panic message displayed on the system console	10-2
Step 2: Categorize the panic message	10-2
Step 3a: Hardware Failure Recovery	10-2
Step 3b: File system problem recovery:	10-3
Step 3c: LAN communication problem	10-4
Step 3d: LVM-related Problem	10-4
Step 3e: Recovery from other situations	10-5
Step 4: Rebooting your system	10-5
Step 5: Monitor the system closely for a while	10-6
A. Using the <code>fsck</code> Command	
How <code>fsck</code> Handles Inconsistencies	A-4
Initialization Phase Errors	A-4
Phase 1 Errors: Check Blocks and Sizes	A-7
Phase 1b: Rescan for More Dups	A-11
Phase 2: Check Path Names	A-11
Phase 3: Check Connectivity	A-17
Phase 4: Check Reference Counts	A-18
Phase 5: Check Cylinder Groups	A-21
Phase 6: Salvage Cylinder Groups	A-22
Cleanup	A-23

Glossary

Index

Figures

2-1. Line Printer Spooling System “Plumbing Diagram.”	2-2
3-1. An example UUCP configuration	3-4
3-2. Testing UUCP related hardware.	3-10
3-3. Testing UUCP’s Devices file.	3-13
3-4. Testing UUCP’s Systems file.	3-16
3-5. Flow of information through UUCP.	3-17
3-6. UUCP “Behind the scenes.”	3-18
6-1. Disk Block/Fragments	6-2
6-2. Disk Block Types	6-3
7-1. Disk Drive Section Map	7-2
8-1. The pvcreate Command Transforms Disks into Physical Volumes	8-2
8-2. The System’s Disk Drives	8-32
8-3. The System’s Disk Drives	8-46
8-4. The System’s Volume Groups and Logical Volumes	8-48

Tables

5-1. System Boot-up Problems (Quick Reference)	5-2
5-2. Files containing boot information for mkboot	5-34
8-1. Sample LVM System Configuration (Part 1)	8-29

Solving Your Problems

We're sorry you're having to read this manual because it probably means that you are experiencing a problem with your HP-UX system. We hope that this manual will help you solve the problem.

It would be impossible to cover every problem you could encounter. Our approach therefore is not to create an exhaustive list of problem and cause descriptions, but rather to provide you information and direction so that you may solve the problem yourself. Where it is appropriate, we will identify common problems and their associated possible causes.

This first chapter is directed at those of you who are new to troubleshooting, although others may want to read it. The chapter covers general troubleshooting techniques (problem definition, problem isolation, record keeping, resources) which can be applied to nearly any subject area. Subsequent chapters discuss how these troubleshooting techniques can be applied in key areas of HP-UX.

The activities described here are formal (for example, creating a list of possible causes and testing each possible cause to see if it is the *actual cause*). For minor problems you encounter, you perform many of them automatically. It is good practice to consciously think about each activity, even for small problems. This will help you to apply the process when you tackle more difficult problems.

Recording the Events

Why keep records?

The first thing you should do when you encounter a problem with HP-UX is to record some basic information about it in a log book.

There are good reasons to keep records of your troubleshooting activities.

- If you encounter the problem again at a later time you may not remember what you previously did to correct it. If you have recorded the problem description and what you did to correct it, you won't have to troubleshoot the problem a second time.
- If you are not around and someone else encounters the problem they may benefit from your notes, especially if the notes are easily accessible. Many people set up system log books to record this information.
- The most important reason to keep records of the problems you encounter (and how you attempted to fix them) occurs when you need to get additional help. This information will be extremely valuable to those attempting to help you.

What information you should record

We encourage you to keep the information shown on the form on the next page. You may want to keep additional information that suits your needs. It may seem like a lot of information to record, but the time you spend recording it now can save you even more time later on. One of the most important pieces of information to record is the HP-UX version number. To obtain this number execute the command:

```
uname -r
```

For information on how to record your problem description, refer to the section called "Defining the problem" later in this chapter. For information on how to determine possible causes, refer to the section "Determining the cause" later in this chapter.

Problem Log Entry:

Date:_____ Time:_____

HP-UX Version Number:_____

Software Version Number(s), if appropriate:_____

Problem Description:_____

Possible Cause:_____

Test performed (to accept/reject this as the *actual* cause):_____

Possible Cause:_____

Test performed (to accept/reject this as the *actual* cause):_____

Problem Resolution:_____

Tools to help you record the information

There are tools that can help you record information (and log files that record certain information automatically). Log files will be covered in this manual where it is appropriate to use them. A tool that records (in a file) the output displayed on your terminal screen is called a *script*(1).

Note

Log files and online records work well *if* your system is up and running and you can get to them. We recommend that you maintain a physical notebook (hardcopies of your online records) so that if your system crashes or the file system containing your records is destroyed, you will still have access to what you've done and what the results were.

Defining the Problem

One of the most important troubleshooting steps is problem definition. You should have a clear understanding of the problem before attempting to solve it. This will keep you focused on that problem and prevent you from going off on tangents. You may even want to write down the problem definition.

When you define the problem, evaluate it in *two* ways: what is going wrong *and* what isn't going right. This subtle distinction will provide you better perspective of your problem. For example:

Suppose that you try to print a file and it doesn't print. What is going wrong is that your output doesn't print or an error message is displayed (or both). Or perhaps garbage is printed instead of your data. What isn't going right is that your output never successfully prints.

It may seem silly to evaluate the problem both ways but you can get different information by doing so. The additional information you get from this double perspective often yields a greater variety of possible solutions to the problem which, in turn, can lead to a *better* actual solution.

You should also consider what you were trying to accomplish when this problem occurred. There might be an alternate way to achieve your goal, eliminating the need to troubleshoot the problem.

Here is a list of things to consider as you define your problem:

- What were you trying to accomplish (end result)?
- What command/operation were you using?
- Have you ever been successful doing this operation before?
 - if *yes*: What things in your system have changed since your previous success (consider *only* those things that are likely to affect what your trying to accomplish)?
 - if *no*: Are there other ways to achieve the same end result
- What are the *symptoms* of the problem? Be as specific as possible in your description(s).

Isolating the Problem

Once you have clearly defined what the problem *is*, one of the best ways to troubleshoot it is to identify what it *isn't*. Eliminate as many unrelated pieces of information and unrelated conditions as possible. For example:

If you are driving along and see smoke coming out from under the hood of your car, the fact that your tires may be a little low on air pressure is probably unrelated. Therefore, in troubleshooting the problem, you need not check the air pressure in your tires.

Here's another example (from HP-UX):

You've just added two new terminals to your system. They are the same model as the others that you've had for a while. An application program that runs fine on all of the previously existing terminals doesn't run properly on the two new terminals. It is unlikely that the problem is with the program; therefore, you need not check for a bug in the code as the reason for the problem.

Eliminate unrelated pieces of information to make the problem more manageable. If it becomes necessary to get someone to help you with the problem, it will be easier to describe the problem to them and easier for them to understand what the problem is.

Problem isolation is done by identifying and eliminating details and facts that you *know* are not related to the problem. If you are not sure whether something is related to your problem or not, don't eliminate it from further consideration.

Suppose, in the above example, you installed an additional disk drive in addition to the two new terminals (to provide for extra file storage). You did not modify the kernel to do this as the I/O structure for it was already built into the existing kernel. You also backed up the system right before installing the new disk drive.

If you have done system backups using the same methods many times before and the backups have never affected the running of your application in the past, eliminate the backup as a possible cause of your problem.

Once you have narrowed the scope of the problem, you can begin to test the items you did not eliminate.

At first you may not be sure whether or not the new disk drive is causing the problem (it is a possible cause). You can test this by disconnecting the drive and attempting to run the application on the new terminals. If the symptoms do not change (the new terminals still do not work but the old ones do), you can eliminate the installation of the disk drive as the cause of the problem. Record this “test” and the results in your log.

This problem isolation process also works well when others come to you with problems. Ask them questions which require specific answers so that you get facts and information directly related to the problem and minimize extraneous information. Suggest things that the person with the problem can test to gain more information about the source of the problem.

Determining the Cause

To determine the *actual* cause of your problem, identify a variety of *possible* causes.

Once you have a list of possible causes, test each of them to verify whether or not it is the actual cause. This is the critical step. The key is to design tests that can test each possible cause individually and provide you with results that allow you to either accept or reject each possible cause. A fairly reliable way to create these tests is to ask the question “if this is the actual cause of my problem, what will I have to do to correct or work around it?”

If, after testing each of your possible causes, you have identified more than one as being the “actual” cause, you may need to develop a new set of tests to eliminate the imposters.

Note

It is possible for a problem to have more than one cause. If you suspect that this is the case, you will need to take corrective action for each of the causes.

Once you have determined the actual cause of the problem, your troubleshooting is complete. You've "shot the trouble." All that you need to do is to take appropriate corrective actions.

The remaining chapters in this manual present examples of how this process works in key areas of HP-UX. The following areas are represented in this manual:

- Line Printer Spooling System Problems
- UUCP Problems
- HP-UX Cluster Problems
- System Boot-up Problems
- File System Problems
- Disk Space Problems
- Problems with Terminals
- System Panics

If the problem you are experiencing is *not* in one of these areas, you may still want to reference the chapter most closely relating to the type of problem you are having. The methods discussed in that chapter (when used with other documentation such as the system administration manuals) might help you locate the cause of your problem.

Preparing for Problems in Advance

Problems cannot *always* be avoided, but you can sometimes take preventive measures. And there are things you can do to lessen the impact of problems when they do arise.

Preventive measures are those things you do to *avoid* problems. These would include things like organizing disk usage to avoid filling up file systems, and hardware preventive maintenance.

When problems do occur, there are usually two things you want to minimize the loss of time and data (both of these translate to money). Careful planning and preparation help here.

Maintaining backup copies of important data is essential in preventing data loss in the event of problems causing corruption of your primary data. The type and frequency of your backups depends on your specific needs. The key question to ask to determine how often you should back up your data is “how much data can you afford to lose?” For help in determining this, see “Backing Up and Restoring Your Data” in the *System Administration Tasks* manual. In critical situations, you might also want to keep extra equipment on hand in case of failure.

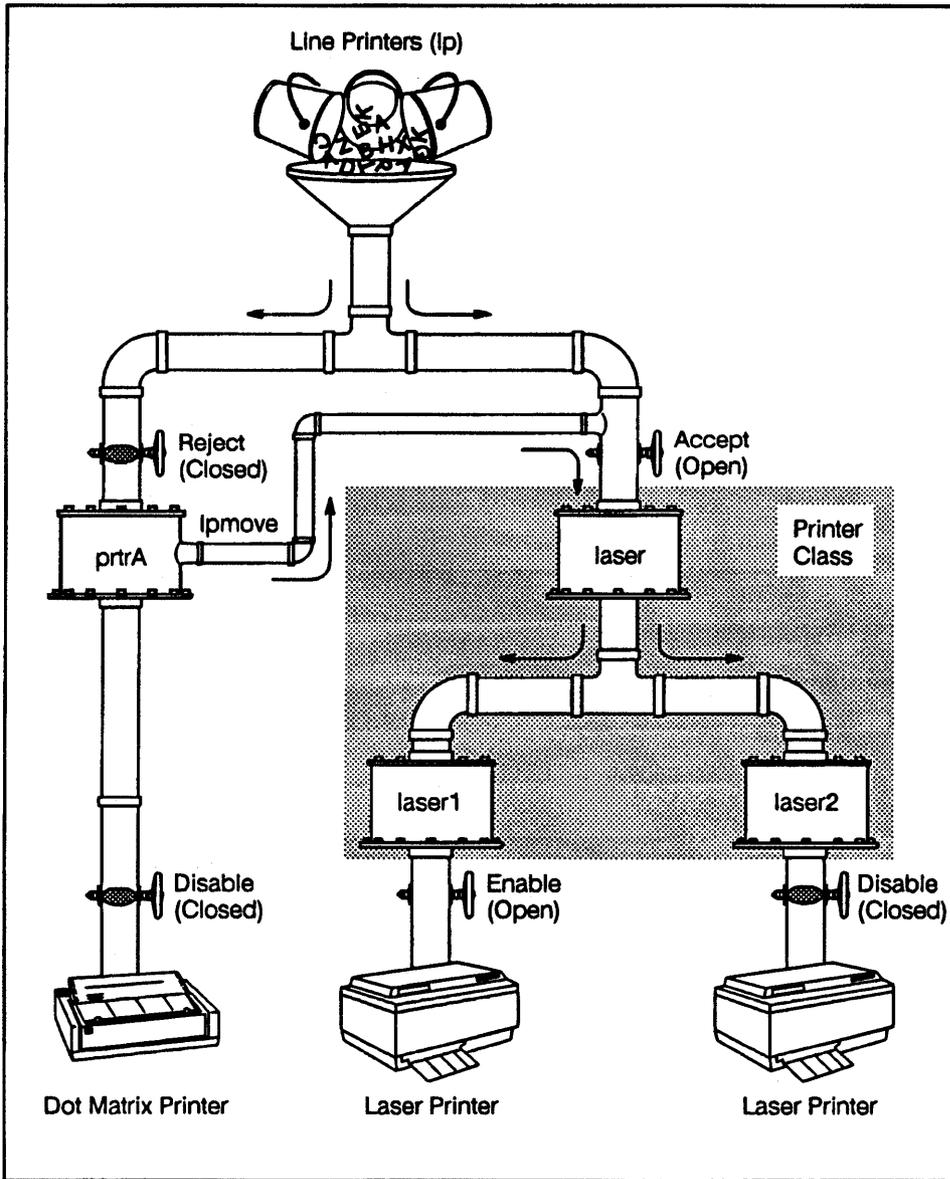
Line Printer Spooling Problems

An overview of the Line Printer Spooling System

The “Line Printer Spooling System” is a set of programs, shell scripts, and directories that control your printers and the flow of data going to them. It helps prevent intermixed listings, provides control of printout routing, and allows users to cancel or restart print jobs.

There are several places where problems can occur in this fairly complex system. To effectively troubleshoot those problems, you should understand the various components of the line printer spooling system and the flow of data through it.

You can think of the line printer spooler as if it were a plumbing system. Figure 2-1 shows how this “plumbing system” might look. The data to be printed represents the “water” in this system. There are various **request directories** that serve as temporary holding tanks. The **accept/reject** and **enable/disable** commands control the flow of data through the *spooling system*, as valves would in a real plumbing system. Interface (shell) scripts near the end of the data flow serve as pumps, which “pump” an orderly flow of data to the printers.



LG200139_001a

Figure 2-1. Line Printer Spooling System "Plumbing Diagram."

2-2 Line Printer Spooling Problems

There is a scheduler (`lpsched`), which controls the routing of print jobs to the printers. It functions as a flow controller in the “plumbing system” to prevent multiple jobs from printing on a printer simultaneously and to provide efficient use of the printers on your system.

If you need to add a printer to your system (or remove one from your system), you can use the “pipe wrench” called the `lpadmin` command to perform the task.

If the “drain gets clogged” for one printer, you can reroute the data for that printer to another by using the `lpmove` command (assuming you have more than one printer), or you can “flush” unwanted data from the spooling system using the `cancel` command.

When you use *remote spooling*, a special shell script (“pump”) is used to send the data to a remote system (via the `rlp` command). A program on the remote system, called `rlpdaemon`, receives the data, directing it into that system’s spooler. Likewise, `rlpdaemon` could be running on your system to receive requests from remote systems.

For a more detailed description of the spooling system and its commands, refer to the *System Administration Tasks* manual under the chapter called “Managing Printer Output”.

Using the LP Spooler with HP-UX Clusters

The Line Printer Spooler can only be run on the cluster server (that is, the line printer scheduler (`lpsched`) can only be executed on the server). The clients can *use* the spooler via the `lp` command. No special configuration is necessary (you do not need to use the remote spooling commands just to access devices on the cluster server). If any `lp` commands (which are valid for use on a cluster client node) fail, the cause is most likely a networking problem.

The following commands are context-dependent files with one element (context) for the server (local root) and one element for all of the clients (remoteroot):

- `/usr/bin/cancel`
- `/usr/bin/slp`
- `/usr/lib/lpsched`
- `/usr/bin/disable`

The Problem Areas

Common problems include:

- Problems with spooler startup.
- Paper jams and printers out of paper.
- You begin printing the wrong printout and need to cancel the printing of the incorrect data.
- Your printout won't print out.
- You might have problems with remote spooling even when local spooling on the system with the printer(s) is functioning properly.

Spooler Scheduler

`/usr/spool/lp/SCHEDLOCK` is created at spooler startup (`lpshed`) to ensure that only one spooler scheduler is running. `lpshed` will not start the scheduler if the file `/usr/spool/lp/SCHEDLOCK` exists. If for some reason the system crashes, it is possible that the system administrator will have to manually remove `SCHEDLOCK` to restart the spooler. The `SCHEDLOCK` file is automatically removed when the spooler is properly shut down (using the `lpshut` command).

Restarting printouts following a paper jam or paper out

Restarting printing from the beginning of the job

When a paper jam occurs in a printer, usually a part of your printout is destroyed and you need to print the output again *from the beginning*. When this happens, do the following:

1. Be sure the printer is in an *offline* mode. (See your printer's owner's manual for information on how to check this.)
2. Issue a `disable` command (at the terminal) to tell the *spooling system* the printer is disabled. Do this *while the printer is offline*.
3. Rethread the paper according to your printer's owner's manual.
4. Place the printer in an *online* mode. (Check your printer's owner's manual for information on how to do this.)
5. *After* putting the printer back on line, issue an `enable` command (at the terminal) to tell the *spooling system* that the printer is ready to begin printing again.

Here's an example showing a printer that jams in the middle of printing (it is configured into the spooling system as "laserjet"):

```
lp -dlaserjet longprintout
```

```
⋮
```

Now the printout jams in the printer. Verify that the printer is offline

```
disable laserjet
```

Put the new paper in the printer and put the printer back on line

```
enable laserjet
```

Note

Under certain circumstances, the printer is automatically disabled for you but, to be safe, it's best to issue the `disable` command (it causes no harm to `disable` an already disabled printer).

Once the printer is re-enabled, the printout in progress will begin printing *from the beginning*.

Restarting printing from the point where it stopped

To continue printing where the job left off, simply correct the paper jam or paper out condition and place the printer back on line. If printing doesn't resume, issue an `enable` command to tell the *spooling system* to resume printing on that printer. If the `enable` command doesn't cause the printing to resume, refer to the section, later in this chapter, called "What to do when the printer won't print".

Canceling a runaway printout

You might discover that the output being printed isn't what you wanted. If you wish to prematurely terminate a printout, issue a `cancel` command. For example:

```
cancel laserjet-1092
```

To do this you must first know the **request-id** associated with the job you want to cancel. The request-id was the number returned by the `lp` command when you started the printing job. For example:

```
lp /etc/motd                print the message of the day
request id is laserjet-5554 (1 file)
```

... yields the request-id "laserjet-5554".

If the request-id has scrolled off the screen or was never printed:

1. issue an `lpstat -u` command to list current print jobs
2. look at the entries with the "on *printername*" at the far right. These are the entries *now printing* on your various printers. If there are more than one of these, use the user name (the second field) and, if necessary, the printer name ("on ... ") to identify which job is yours. Once you identify the entry that represents your job, you will find its request-id in the first field. The `lpstat -u` command's output will look similar to this:

```
laserjet-5557      opr          207253    Jul 26 16:25 on laserjet
laserjet-5558      root         1766     Jul 26 16:26
```

Note It is also possible to cancel a job that has not yet started printing using `cancel`.

Moving print requests from one printer to another

If you have more than one printer and one of them stops functioning, you can transfer the jobs that are waiting to be printed on the non-functional printer to a printer that is working. Use the command `lpmove` to do this. The `lpmove` command will *not* work while the line printer scheduler is running, so you must first shut down the scheduler. Perform the following steps:

1. Shut down the line printer scheduler, using the command `/usr/lib/lpshut` (unless you've changed `lpshut`'s permissions, you will need to be a superuser to issue this command).
2. Issue an `lpmove` command to move requests for the disabled printer to a working printer's request queue (the request currently printing and those waiting to be printed will be moved). `lpmove` will also issue a `reject` command to the disabled printer's queue to prevent further print requests from being scheduled for that printer.
3. Issue a `/usr/lib/lpsched` command to restart the scheduler (as with step 1, you will probably need to be a superuser to issue this command).

Note All printers that are printing at the time the lp scheduler is shut down will stop printing. Requests that were printing at that time will be completely reprinted after the lp scheduler is restarted.

Example (Using lpmove)

In this example, printer “laserjet” develops a problem that prevents it from printing. There is a job (request-id laserjet-7342) currently printing and there are two print jobs (request-id’s laserjet-7343 and laserjet-7344) waiting to print. A second printer (lj2000) has been configured into the spooling system. Perform the following steps to move the printing requests from printer “laserjet” to printer “lj2000”:

```
lpstat -u      here are the print requests before the lpmove is done
laserjet-7342   judyg      630629   Jul 28 16:47 on laserjet
laserjet-7343   daleb     328922   Jul 28 16:51
laserjet-7344   daleb     1766     Jul 28 16:52
lj2000-5726    petern    10528    Jul 28 16:53 on lj2000
```

```
/usr/lib/lpshut                shut down the lp scheduler
/usr/lib/lpmove laserjet lj2000 move the print jobs
/usr/lib/lpsched              restart the lp scheduler
```

```
lpstat -u      here are the print requests after the lp move is done
lj2000-5726     petern    10528    Jul 28 16:53 on lj2000
lj2000-5727     judyg     630629   Jul 28 16:47
lj2000-5728     daleb     328922   Jul 28 16:51
lj2000-5729     daleb     1766     Jul 28 16:52
```

Note It is also possible to move individual requests using the `lpmove` command. To do this, replace the source (first) printer’s name in the `lpmove` command with the request-id. It will still be necessary to shut down the scheduler to do this. If you move *individual requests*, `lpmove` will *not* issue the `reject` command to the printer.

What to do when the printer won't print

Sometimes, the most difficult line printer problems to track down are those in which a printer won't print. The difficulty comes from the number of possible causes of this condition. The data to be printed on the paper must travel through a lot of "plumbing" (refer again to figure Figure 2-1). If there is a clog *anywhere* along the way, the data will not reach your printer. Even if the data reaches your printer, there are a several possible reasons why the printer won't print it.

We'll discuss most of the likely causes here using the methods of troubleshooting outlined in chapter 1 of this manual.

Defining the problem:

Include the following things when you are defining your problem:

- If you have more than one printer, which printer won't print?
- Are other printers printing?
- Has this printer ever printed before on this system or are you trying to get it going for the first time?
- Were any error messages associated with the failure to print (write them down)?
- Can others print successfully on this printer?

Isolating the problem:

If the problem does not affect every printer (or your printer is the only one on the system) you will need to consider possible causes associated with only one printer (see the next section "Determining the cause:").

If the problem affects all printers on your system (or your printer is the only one on the system), you need to consider possible causes that are global (see the next section "Determining the cause:").

If others on your system can print and/or you can successfully print other output/files, check the following:

- Be sure that the user "lp" has permission to access the file you are trying to print (the file needs to have read access set for "the world"). If you need to

print a file with restrictive permissions, use the `cat` command and pipe the output of `cat` into the program `lp`. For example, suppose you need to print a file called `/users/secret` that has the following permissions:

```
-rw-----  1 root    other      5933 Aug  2 14:48 /users/secret
```

You could use the following command (which will eliminate the need for you to change the file's permissions, just so that you can print it):

```
cat /users/secret|lp
```

- Issue the command `lpstat -t` and compare the priority of your print request (the one associated with the request-id that was returned to you when you issued the `lp` command). Be sure that the priority of the print request is at least as high as its printer's **priority fence**. If your print request has a *lower* priority than its printer's priority fence, *it will not print*. Use the command `lpalt` to set the priority of the print request so that it is equal to or greater than the priority fence for its printer.
- Occasionally (but rarely) the line printer scheduler (`lpsched`) gets out of sync in printing requests. When this happens, you can usually fix the problem by shutting down the scheduler (`/usr/lib/lpshut`) and restarting it (`/usr/lib/lpsched`).

Note When you shut down the scheduler, any jobs that are printing, *on any printer in your system*, will be terminated. These jobs will be printed in their entirety when you restart the scheduler.

Determining the cause:

Once you have determined the scope of the problem, create a list of possible causes (you might want to refer to the "plumbing diagram" when creating your list). Here are some possible causes to get you started.

For problems associated with only one printer. Check the following:

- Printer is powered on and is online (see your printer's owner's manual for information on how to check this)

2-10 Line Printer Spooling Problems

- Printer has paper
- Data cable is connected
- Printer is enabled (check using the command `lpstat -p`)
- Printer is accepting requests (check using the command `lpstat -a`)
- The printer is defined within the line printer spooling system
 1. Issue the command `lpstat -s`
 2. Look at the printer names (first field of each entry listed)
 3. If the printer name you specified as the destination in your print request is listed, that printer is defined within your spooling system.
 4. If you did *not* specify a printer name in your print request, the system will use the printer identified as the **System Default Destination**
- Correct device file is associated with the printer you specified
 1. Issue the command `lpstat -v`
 2. Locate the entry corresponding to your printer name (“Device for ... ”)
 3. Locate the name of the device file corresponding to your printer (last field of each entry) and use the `cat` command to send a file to this device file.

For example:

```

lpstat -v
device for wisdom: /dev/null
device for alttag: /dev/tty0p3
device for laser: /dev/tty0p3
device for lp: /dev/tty0p4
device for quiet: /dev/tty0p3
device for taglp: /dev/tty0p4

```

```

cat /etc/motd > /dev/tty0p4

```

In the above example we attempted to send the “message of the day” (file `/etc/motd`) to the device file associated with our printer called “lp”. If the file prints on the printer, then the device file is set up to point to the correct hardware address (and the printer is physically functioning properly). If the file *doesn't* print, there might be a problem with the device file; it might be pointing to the wrong hardware address. See the next step for more information on how to check this.

4. Look at the device file corresponding to your printer in its entry and be sure that this device file matches the hardware location of your printer.
 - a. *For Series 300 Computers:* Look at the minor number for this device file using the `ll` command. Be sure the select code (first two hex digits following the “0x”) corresponds to the card your printer is plugged into. If you have a serial printer hooked to a MUX, you will also need to check the port number (next two hex digits) to be sure that it matches the port where your printer is attached.
 - b. *For Series 800 Computers:* Look at the hardware address for this device file using the command `lssf {your-device-file-name}` and verify that the hardware address listed for the device file matches where your printer is attached. If you have a serial printer attached to a MUX, also verify that the port number listed by the `lssf` command matches the port where your printer is plugged in.
- You specified the correct printer (using the `-d` option of the `lp` command)
- System default destination is set for the correct printer (you can check this by using `lpstat -s` and you can set the system default destination using the `-d` option of the `lpadmin` command)

For problems of a global nature. Check the following:

- The line printer scheduler is running (check using the command `lpstat -r`)

Where to go for more information

- For information about setting up and using the Line Printer Spooling System, see “Managing Printers and Printer Output” in the *System Administration Tasks* manual.
- For information about installing a printer on your computer, see *Installing Peripherals*.

UUCP Problems

An Overview of UUCP

UUCP (upper-case) is an acronym for “Unix to Unix CoPy.” It is a complex set of configuration files, shell scripts and executable programs that enable you to perform “network” operations using serial lines.

As with most networking subsystems, UUCP offers three basic services; remote login, file transfer, and remote command execution. The commands that support these are `cu`, `uucp`(lower case), and `uux` respectively. These commands rely on the configuration files, shell scripts and executable programs to perform the three services for you.

Several versions of UUCP exist. The different versions perform the same functions, but differ in their underlying configuration files etc. We will discuss troubleshooting with respect to the *HoneyDanBer* version of UUCP (the version running on HP-UX, first implemented on the Series 300 at release 6.2 and on the Series 800 beginning with release 2.0).

The Problem Areas

Because of its complexity, there are many things that can go wrong with UUCP and troubleshooting them can get tricky. There are tools to help you and we will discuss a “layered” method of troubleshooting in this chapter, which should help you find most of the problems.

File transfer and remote command execution (using `uucp` and `uux`) do not occur immediately. There are background processes that must create files containing instructions and data, establish a remote connection, and login into the remote system. These background processes might not be running at the time of your request and you might have to wait until they run again for the

transfer to take place or the remote command to be executed. The point is that just because the activity you requested doesn't happen immediately does not necessarily mean that anything is wrong. Run the following command to force immediate execution of the UUCP background programs:

```
uucico -r1
```

Executing `uucico` directly should *only* be done when you are debugging a problem. It is normally scheduled by other processes such as the `cron` daemon.

Many UUCP problems are related to errors in configuration files. An HP-UX tool called SAM provides you with a menu-driven interface for setting up UUCP. SAM will create many of the configuration files for you, reducing your chances of introducing errors in these files. SAM will set up the configuration files in a very specific (but valid) way. You can manually customize the files, once SAM has created them, if you really need to so. For information on SAM and its capabilities, consult your *HP/9000 System Administration Tasks Manual*. We strongly recommend that you use SAM to set up UUCP. Problems can also occur in the following areas:

- Hardware Problem (component failure)
- Hardware Problem (incorrect cable wiring)
- Hardware Problem (incorrect modem configuration settings)
- Hardware Problem (bad connection causing incorrect transmission of data)
- Hardware Problem (someone is already using it)
- Configuration Problem (wrong password, system name, etc. for remote system)
- Disk Space Problem (file cleanup should be performed regularly)
- File Permissions Problem (one or more of the UUCP files or the files/directories being operated on do not have the proper access permissions)

Because of the large number of potential problems, it is helpful to have a systematic approach to troubleshooting them.

It is also helpful to have an example of a working configuration to compare yours against. Here is an example of a three-computer uucp configuration. The following information is provided for each of the systems in the configuration.

3-2 UUCP Problems

- Device file descriptions (which names are used and their purpose).
- Device file configuration information (output of `ll` (showing permissions/ownership, major and minor numbers), `lssf` (S800s only, showing hardware addresses and drivers used).
- Modem phone numbers (if appropriate). Where supplied, these numbers are hypothetical.
- Entries for the `/etc/passwd` file.
- Entries for the `/etc/inittab` file.
- Entries for the `/usr/lib/uucp/Devices` file.
- Entries for the `/usr/lib/uucp/Systems` file.

Examples of a modem connection and a hardwired (direct) connection are provided in this configuration. Two of the systems are 800 Series computers and one is a Model 350. Figure 3-1 (next page) shows how they're connected.

An example of a UUCP configuration

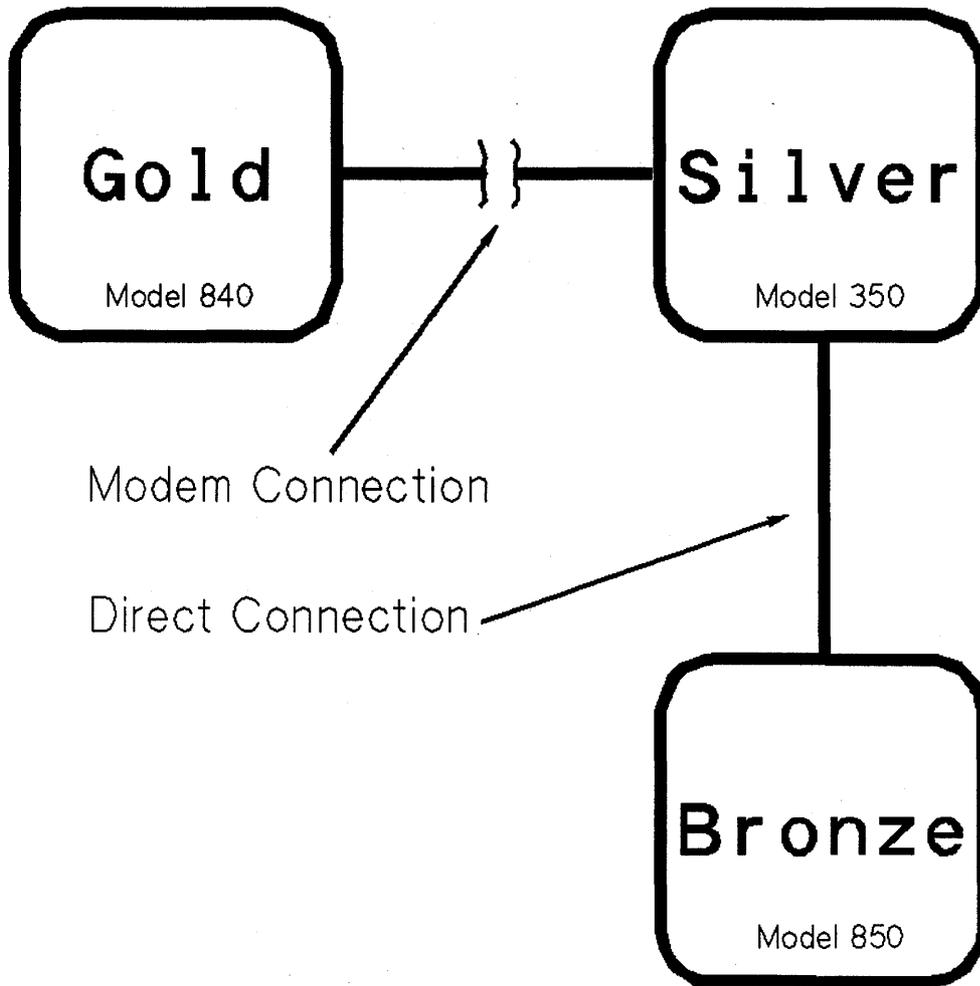


Figure 3-1. An example UUCP configuration

Note

In the listings of system information below, indented lines represent a continuation of their previous line.

SYSTEM #1 System Name: Gold (System Model: HP9000/840)

Device Files Used:

```
/dev/cuaOp1    for dialing (modem control)
/dev/culOp1    for outbound communications
/dev/ttydOp1   for inbound communications
```

```
crw-rw-rw-  1 root  other  1 0x000001 Jun 21 16:43 /dev/cuaOp1
crw-rw-rw-  1 root  other  1 0x100001 Jun 21 16:42 /dev/culOp1
crw--w--w-  1 root  other  1 0x200001 Jun 21 16:42 /dev/ttydOp1
```

```
mux0 lu 0 port 1 hardwired address 8.1 /dev/cuaOp1
mux0 lu 0 port 1 callout address 8.1 /dev/culOp1
mux0 lu 0 port 1 callin address 8.1 /dev/ttydOp1
```

Modem Phone Number:

555-1234

Entries in /etc/passwd

```
uucp:7Rpmzdxj0nNYk:5:1:uucp:/usr/spool/uucppublic:
    /usr/lib/uucp/uucico
nuucp:7Rpmzdxj0nNYk:5:1:uucp:/usr/spool/uucppublic:
    /usr/lib/uucp/uucico
```

Entry in /etc/inittab

```
u0:2:respawn:/etc/getty -h -t 165 ttydOp1
```

Entries in /usr/lib/uucp/Systems

```
modem Any ModemOp1 2400 -
silver Any;5 ACU,g 2400 555-6789 "" \r\d\r\d\r ogin:-BREAK-ogin:
    uucp assword: sss123
```

Entries in /usr/lib/uucp/Devices

Modem0p1 cua0p1- 2400 direct *Not set up by SAM. Used for sending
commands directly to the modem.*

Direct cua0p1 - 2400 direct

ACU cul0p1 cua0p1 2400 hayes

SYSTEM #2 System Name: Silver (System Model: HP9000/350)**Device Files Used:**

/dev/tty02 *normal device file for the mux port*

/dev/cua01 *for dialing (modem control)*

/dev/cul01 *for outbound communications*

/dev/ttyd01 *for inbound communications*

crw-rw-rw- 1 root other 1 0x0d0004 Jun 21 16:43 /dev/cua01

crw-rw-rw- 1 root other 1 0x0d0001 Jun 21 16:42 /dev/cul01

crw-rw-rw- 1 root other 1 0x0c0004 Jun 21 16:46 /dev/tty02

crw--w--w- 1 root other 1 0x0d0000 Jun 21 16:42 /dev/ttyd01

Modem Phone Number:

555-6789

Entries in /etc/passwd

uucp:5ZrP/ZsZ/nVY2:5:1:uucp:/usr/spool/uucppublic:
/usr/lib/uucp/uucico

nuucp:5ZrP/ZsZ/nVY2:5:1:uucp:/usr/spool/uucppublic:
/usr/lib/uucp/uucico

Entry in /etc/inittab

u0:2:respawn:/etc/getty -h -t 165 ttyd01

Entries in /usr/lib/uucp/Systems

modem Any Modem01 2400 -

gold Any;5 ACU,g 2400 555-1234 "" \r\d\r\d\r ogin:-BREAK-ogin:

3-6 UUCP Problems

```
uucp assword: ggg123
bronze Any;5 bronze,f 9600 - "" \r\d\r\d\r ogin:-BREAK-ogin:
uucp assword: bbb123
```

Entries in /usr/lib/uucp/Devices

```
Modem01 cua01 2400 direct      Not set up by SAM. Used for sending
                                commands directly to the modem

ACU cul01 cua01 2400 hayes
Direct cua01 - 2400 direct
bronze tty02 - 9600 direct
```

3

SYSTEM #3 System Name: Bronze (System Model: HP9000/850)

Device Files Used:

```
/dev/ttyOp2    for outbound communications
/dev/ttydOp2   for inbound communications

crw--w--w-  1 root  root    1 0x000002 Jun 21 11:31 /dev/ttyOp2
crw--w--w-  1 root  users  1 0x200002 Jun 14 10:27 /dev/ttydOp2

mux0 lu 0 port 2 hardwired address 2.4.1 /dev/ttyOp2
mux0 lu 0 port 2 callin address 2.4.1 /dev/ttydOp2
```

Entries in /etc/passwd

```
uucp:hc0ae1Z8GoHQg:5:1:uucp:/usr/spool/uucppublic:
    /usr/lib/uucp/uucico
nuucp:hc0ae1Z8GoHQg:5:1:uucp:/usr/spool/uucppublic:
    /usr/lib/uucp/uucico
```

Entry in /etc/inittab

```
u0:2:respawn:/etc/getty -h -t 165 ttydOp2
```

Entries in /usr/lib/uucp/Systems

```
silver Any;5 silver,f 9600 - "" \r\d\r\d\r ogin:-BREAK-ogin:
uucp assword: sss123
```

Entries in /usr/lib/uucp/Devices

```
silver ttyOp2 - 9600 direct
```

Defining the problem:

Before attempting to solve your problem, answer the following questions.

- What operation were you trying to perform when the problem occurred (remote login, file transfer, remote command execution)?
- Have you ever been successful performing this operation in the past (that is, did this *stop* working or *has it ever* worked)?
- Does this problem occur between your system and *only one* remote system or does it happen between your system and *more than one* remote system?
- Do remote systems have problems contacting your system, too?

Keep the answers to these questions in mind as you go through the following steps.

Isolating the problem

To isolate the problem, begin at the lowest level of communication possible and add communication layers as long as your results are successful. When you detect a failure, look at the files, scripts and programs associated only with the layer you most recently added.

Step 1: Test the hardware

This first step bypasses UUCP altogether. You simply test the integrity of your device file (and the hardware connection).

Modem Connections

For a modem connection, use the `cat` command (as shown in Figure 3-2) to allow you to send characters from your keyboard through the device file to the

communication port your modem is connected to (use the name of your device file). Use the device file with the name that begins with “cua” (for example `/dev/cua01`). Once you have executed this command, all further keystrokes from your keyboard will be sent to the device file until you type `CTRL-d` (or whatever you’ve defined as your *eof* character). Do not be concerned if your modem doesn’t act upon any modem commands you send it. The only thing to be concerned about at this time is that the *transmit data* and *receive data* indicators blink when you type characters.

In our example configuration, if we were testing a problem on system “gold”, we would enter the command (on that system):

```
cat > /dev/tty0p1
```

Then we would type characters on the keyboard while watching the modem *transmit data* and *receive data* indicators to see if they blink. When we have observed whether or not the indicators blink, we would end the `cat` command and get a new shell prompt by typing a `ctrl-d` character.

Command	Files Involved
<pre>cat > /dev/{your device file} or kermit -l {line} -b {speed} -c</pre>	<div data-bbox="705 819 1065 895" style="border: 1px solid black; padding: 5px; display: inline-block;"><pre>/dev/{your device file}</pre></div>

Figure 3-2. Testing UUCP related hardware.

If the modem does not respond, check for the following things:

- The modem is powered on and all appropriate cables are tightly connected.
- The modem is configured properly. Each model/brand of modem has its own configuration switches/commands. In general, configure your modem as follows:
 - Recognize DTR
 - Result Codes are Verbal
 - Enable Result Codes (from commands)
 - Echo characters when in command mode
 - Enable Autoanswer (when used for incoming connections)

- Disable Autoanswer (when used for outgoing connections) *NOTE: some modems have the ability to switch autoanswer modes when they detect a drop of the DTR line.*
- Enable command recognition
- Indicate the presence of a carrier detect signal
- Set the speed (if it is adjustable) to the desired setting
- The cable is connected to the correct port (the one associated with the device file you redirected the output to).
- The cable you are using is of the correct type (see the manual *UUCP, HP-UX Concepts and Tutorials (HP Part No. 97089-90053)* for information on proper selection and wiring for UUCP cables).

Direct Connections

For a **direct connection**, use the `/usr/bin/kermit` command (as shown in Figure 3-2). Substitute in the name of your device file and the appropriate line speed (such as 9600). You can find out the speed to use by executing the command:

```
stty < /dev/your-device-file-name
```

If there is a **getty** running on the remote system's incoming port, you should see a login prompt once **kermit** has connected. If there is not a **getty** running on the remote system's incoming port, you can run **kermit** on that system using the remote system's incoming port. You should then be able to type characters on one system and see them on the other. For a detailed description of **kermit**'s syntax and options, see the *kermit(1)* manual reference page.

On our example system "bronze" we would enter the command

```
kermit -l tty0p2 -s9600
```

If the login message doesn't show up on your local screen or you are not able to send characters between systems with two **kermits** running, then check the following:

- The cable is connected to the correct port on the local system (the one you connected the local **kermit** to)

- The cable is connected to the correct port on the remote system (the one you connected the remote kermit to)
- The cable you are using is of the correct type (see the manual *UUCP, HP-UX Concepts and tutorials (HP Part No. 97089-90053)* for information on proper wiring for UUCP cables).

If the modem responded to your commands or kermit was successful at connecting to the remote system, you can feel confident that the hardware is set up and working properly. You can then proceed to step 2.

Step 2: Use `cu` to test `/usr/lib/uucp/Devices`

In this step, we begin testing the various components of UUCP. Use the `/usr/bin/cu` (Call UNIX) command to perform an operation that will use the `/usr/lib/uucp/Devices` file to specify which device file we want to use, but will *not* use the `/usr/lib/uucp/Systems` file. This will test the integrity of the “Devices” file on your system.

Modem Connections

Use `cu` to call a phone in your office. You can do this with the first command shown in Figure 3-3.

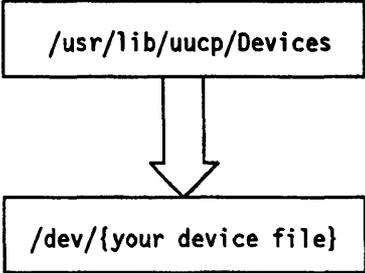
Command	Files Involved
<pre>cu -s [speed] -l [line] [phone-no] or cu -s [speed] -l [line] dir</pre>	 <pre> graph TD A["/usr/lib/uucp/Devices"] --> B["/dev/{your device file}"] </pre>

Figure 3-3. Testing UUCP's Devices file.

If the phone doesn't ring as a result of your issuing the cu command, check the following:

- You typed the correct phone number
- An entry in `/usr/lib/uucp/Devices` for the specified speed and/or line was not found (perhaps it's been commented out)
- The access permissions on the file `/usr/lib/uucp/Devices` would not allow cu to access it.
- An invalid dialer was specified in the `/usr/lib/uucp/Devices` entry.

For a list of standard dialers, check the file `/usr/lib/uucp/Dialers`. There are entries there for commonly used modems. If you are using a modem that is not compatible with any of the entries in the "Dialers" file, you will either need to edit "Dialers" or use the dialit program (in which case

you will have to modify the source code (`dialit.c`) to operate with your modem and compile it to create a new version of `dialit`. For information on modifying “Dialers”, see the manual *UUCP, HP-UX Concepts and Tutorials*.

3

- Someone was already using the modem. You can check to see if anyone is using the modem using the `ps -ef` command or by checking the modem to see if it is currently active.
- A lock file is present in the directory `/usr/spool/uucp` (and no one is using the modem). The lock file will be named `LCK.<your device file name>`.

If you are successful at getting the office phone to ring, try using `cu` to dial the phone number of a remote system. You should get a message that says “CONNECTED.” If not, the problem might be on the remote end (such as the modem doesn’t answer).

Direct Connections

Use `cu` with the `-s` and `-l` options and the `dir` flag as shown in Figure 3-3. For the `line` parameter, use the device file associated with the port being used for UUCP.

On our example system “bronze”, we would enter the command:

```
cu -s9600 -l tty0p2 dir
```

If you do not get the message “CONNECTED”, check the following:

- Make sure you specified `dir` in the command
- An entry in `/usr/lib/uucp/Devices` of the specified speed/line was not found (perhaps it was commented out).
- The access permissions on `/usr/lib/uucp/Devices` would not allow `cu` to access it.
- Someone is already using the connection. Use `ps -ef` to check for any activities associated with the device file you specified in your `cu` command.
- Be sure that the file `/usr/lib/uucp/Devices` exists
- A lock file exists in the directory `/usr/spool/uucp` (and no one is using the connection). The lock file will be named `LCK.<your device file name>`.

Note

Whether using modem or direct connect, once you are successful at getting the “CONNECTED” message, enter  `return` to close the connection. You can now be confident that the problem isn’t with the file `/usr/lib/uucp/Devices`. And, if using a modem, you can be confident that the dialer you specified in `/usr/lib/uucp/Devices` is working.

Step 3: Use `cu` to test `/usr/lib/uucp/Systems`

Once you are confident that the hardware, device file and the file `/usr/lib/uucp/Devices` are properly set up you are ready to test out the integrity of the file `/usr/lib/uucp/Systems`. This file contains entries for remote systems that include the names, valid calling times, phone numbers, login information and login passwords. If a remote system is listed in the “Systems” file, you can use `cu` to call the system without having to specify the speed, line or phone number to use. UUCP will look up this information in `/usr/lib/uucp/Systems` and then reference `/usr/lib/uucp/Devices` and the appropriate device file for you. Enter the command shown in Figure 3-4, substituting the name of a valid system in your “Systems” file.

For our example system “silver”, we could dial our example system “gold” using the command:

```
cu gold
```

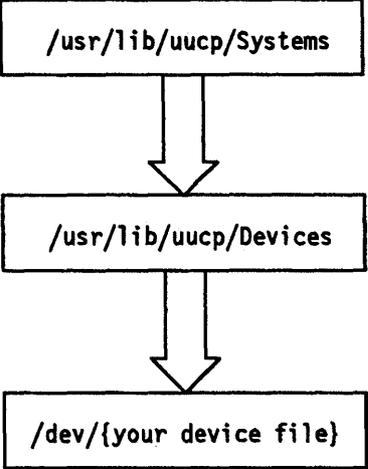
Command	Files Involved
cu system-name	 <pre> graph TD A["/usr/lib/uucp/Systems"] --> B["/usr/lib/uucp/Devices"] B --> C["/dev/{your device file}"] </pre>

Figure 3-4. Testing UUCP's Systems file.

If you are *not* successful at getting the “CONNECTED” message, check for the following things:

- There is an entry in your `/usr/lib/uucp/Systems` file for the system you specified
- The phone number in the “Systems” file entry is correct
- The permissions on `/usr/lib/uucp/Systems` permit `cu` to access it

The “Systems” file entries contain many fields of information; it is easy to make a mistake when entering the information. Again, we suggest that you use the `SAM` utility to configure UUCP as it will create the entries in “Systems” for you, reducing your chances of introducing errors into `/usr/lib/uucp/Systems`.

Once you can successfully connect to a remote system by using the command:

```
cu {system name}
```

you can be confident that your “Systems” file isn’t causing your problem. You are ready to try a uucp file transfer.

Step 4: Use uucp to transfer a file

At this step there are many new things that become involved and it is helpful to take a short “behind the scenes” look at UUCP to understand what types of things can go wrong.

uucp and uux do not actually do the inter-system communication. Instead, they create work files in special directories (located in /usr/spool/uucp) and call upon a process called uucico to perform the actual work. These work files are of several types. Some contain instructions for the uucico processes and some contain actual data to be transferred (or references to it). Figure 3-5 shows the flow of information.

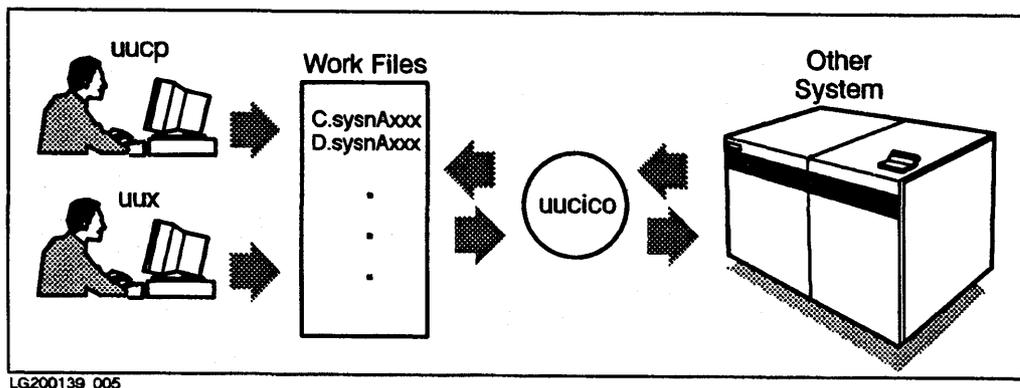


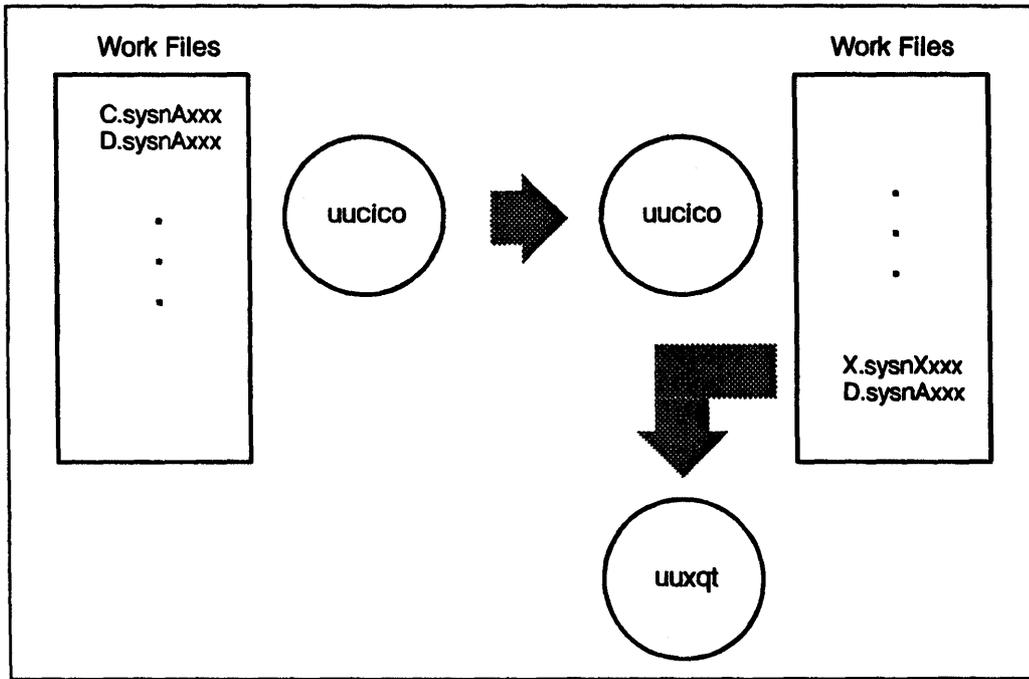
Figure 3-5. Flow of information through UUCP.

uucico on your local system will use the information in the “Systems” and “Devices” files to establish a connection and communicate with uucico on the remote system.

Each type of work file has its own internal format and each has a slightly different format to its name. Files created by uucp get transferred to their ultimate destination. Files created by uux, on the other hand, are placed in the working files directory of their destination system with a slightly altered name (their contents remain unchanged). From there a process called

uuxqt evaluates them and performs the uux requested commands. See figure Figure 3-6.

3



LG200139_006

Figure 3-6. UUCP “Behind the scenes.”

A file called `/usr/lib/uucp/Permissions` controls what a remote system can do on your system and where it can put files. Likewise the remote system’s “Permissions” file can control access to itself by your system. See *UUCP, HP-UX Concepts and Tutorials* for a complete description of the entries in “Permissions.”

Use `uucp` now, to attempt a file transfer to or from a remote system. From our example system “gold”, we could transfer the `/etc/motd` file to system “silver” using the command:

```
uucp /etc/motd silver!/etc/motd
```

If you get an error, try to figure out where in the above flow description the problem might have occurred. Things to look for are:

- Have you supplied a valid system name to `uucp` for the remote system?
- Does the “Permissions” file on the remote system allow you access to the place you requested your file to go? Try sending your file to the `uucppublic` directory on the remote system.
- Can `uucico` be scheduled by `uucp` (or `uux`)? Check the permissions and ownership of `/usr/lib/uucp/uucico`.

Obtaining UUCP status information

The four-step procedure in the previous section should help you resolve most UUCP problems. Due to the complexity of UUCP, there might be problems that can't be located using the above procedure.

Fortunately there are tools and log files that can be used to help you determine what's wrong. Here is a list of tools available and a brief description of them. For more information on them, consult the manual reference pages listed here or the manual *Remote Access: User's Guide*.

Helpful Troubleshooting Tools

Tool/File name	Description	Manual Ref. Pages
/usr/bin/uulog <i>(command)</i>	Searches log files located in /usr/spool/uucp/.Log/uucico and /usr/spool/uucp/.Log/uuxqt directories and displays log entries. Messages displayed to the right of the date field are discussed in the manual <i>Remote Access: User's Guide</i> in the chapter called "uucp Troubleshooting Tools."	<i>uucp(1)</i>
/usr/lib/uucico -r1 -xn (See also uutry) <i>(command)</i>	uucico , the actual file transfer program of the UUCP system can be started manually using its debugging option (-x). A debugging level, specified with the "-x" option, determines how much debugging information is displayed. This can be useful in determining which step in the establishment of communication is failing.	<i>uucico(1m)</i>
/usr/lib/uucp/Uutry <i>(script)</i>	Used to run uucico with debugging output. Yields output that is easier to understand than the output of the " uucico -x " command. Preferred over running " uucico -x " directly.	

3

Helpful Troubleshooting Tools (continued)

Tool/Filename	Description	Manual Ref. Pages
/usr/spool/uucp/DIALLOG (log file)	If you are using the program "dialit" instead of the /usr/lib/uucp/Dialers file to perform the dialing of your modem lines, you can check the file "DIALLOG" for information on the modem used, telephone # dialed, and the result of the dialing (success or failure).	
/usr/bin/uustat (command)	Shows the status of pending uucp and uux commands (those not yet processed by uucico). It can also be used to cancel uucp and uux requests.	<i>uustat</i> (1)
/usr/bin/uuls (command)	Lists the contents of the spool directories. It can be used before and after a uucp or uux command to see if the work files are being created. It can also be used after uucico has been run to verify the successful transfer of workfiles.	<i>uuls</i> (1m)
/usr/lib/uucp/uucheck (command)	Can be used to check for the presence of the UUCP required files and directories. Using the (-v) option, uucheck will also interpret the /usr/lib/uucp/Permissions file entries. This can be helpful when determining capabilities of others dialing into your system, to see if you have them set correctly.	<i>uucheck</i> (1m)

Where to go for more information

- For detailed information on setting up and managing UUCP on your computer, and for information on cable wiring, refer to *Remote Access: User's Guide*.
- For information on using UUCP in an HP-UX Cluster, refer to *Managing Clusters of HP 9000 Computers Sharing the HP-UX File System*.

HP-UX Cluster Problems

An Overview of HP-UX Clusters

An HP-UX cluster is a group of HP9000 computers connected together by a Local Area Network (LAN), sharing a common directory tree. One computer (known as the **cluster server**) has the root file system disk attached to it. The cluster server handles file system I/O requests from the other computers in the cluster via the LAN.

The computers in the cluster (other than the cluster server) are known as **clients**. By default, clients swap to the cluster server's disk (that is, they share the cluster server's swap space). Disks can be attached to clients and used by the clients for local swapping, and they may have local file systems on them. Although the file systems are "local," they are accessible by all of the computers in the cluster.

Although the computers in an HP-UX Cluster share a common file system, they have separately running kernels, each with its own I/O configuration. To handle the differences in configuration between the various computers in a cluster (also known as **cnodes**), Context-Dependent Files are used. **Context-Dependent Files** (also called **CDFs**) are actually special *directories* (known as **hidden directories**), which contain different versions of a particular file's contents, to be used in different *contexts*. When a file is converted to a CDF, its name becomes the name of the hidden directory containing files representing the various contexts for this "file."

Here are a couple of examples to show what is meant by “different contexts.”

EXAMPLE 1

In an HP-UX cluster that contains both Series 300/400 and Series 700 computers, a program that must be run on both types of machines must have two versions of its “executable” file. One version contains the code to run on a 700 Series computer and one contains the code to run on a 300/400 Series Computer. Because it is the same program, it would be best to call it by the same name on both machines. Because both machines share a common file system, there must be a way, not only to have two files of the same name co-exist in the file system, but also to ensure the correct version is executed from each machine. One of the “contexts” that can be used in creating directory entries in a CDF is “processor type”.

EXAMPLE 2

Certain files (such as `/etc/inittab`, `/hp-ux`, and the directory `/dev`) must be customized for each system running in the cluster. The “context” in this case would be the machine name.

The valid **context types** which CDF files can have are:

- cnode name (this is the *name portion* of the ARPA network hostname, see *hostname(1)*)
- floating-point hardware type (HP-MC68881, HP98248A or HP98635A)
- processor architecture (HP-MC68020 or HP-PA)
- file system type (localroot, remoteroot)
- the word “default”

The **context string** of your computer is an ASCII string created by concatenating information from the above categories in a prioritized order. This context string is sent between cnodes as part of HP-UX cluster LAN messages. It is used to search context-dependent files (hidden directories) for the proper entry to use. For a more detailed description of this, see the section called “*What is Context*”, in Chapter 2, “Understanding Clusters” of the manual *Managing Clusters of HP 9000 Computers*.

To see the context string for your computer, enter the command:

```
getcontext
```

This will print the string on your screen (like this):

```
systemmm HP-MC68881 HP-MC68020 remoteroot default
```

During a file search, the cluster server will use the categories of the context string (from left to right) when searching for entries in a hidden directory (until it finds a match or runs out of entries to search).

Note

For the most part, context-dependent files are transparent to the user. HP-UX handles the context searching for you based on your computer’s context string. They are being discussed here because it is necessary to know about them during HP-UX cluster troubleshooting activities. Hidden directories (CDFs) can be identified by a “+” at the end of a file name in the output of many HP-UX commands. The “+” is *not* part of the file’s name; rather it is appended to the file’s name in the command’s output to show that the file is actually a context-dependent file.

A “+” can be appended to a CDF name to escape the CDF mechanism and to see the entries within the hidden directory. For example, if `/hp-ux` was a hidden directory and you wished to see the entries within it, you could use the following `ls` command to view them:

```
ls /hp-ux+    the “+” tells the file system to display the
               contents of the hidden directory /hp-ux
```

Important

Due to their underlying complexity, it is important that you have a thorough understanding of HP-UX cluster operation before attempting to troubleshoot problems in this area.

Extensive information on HP-UX cluster operation is located in *How HP-UX Works: Concepts for the System Administrator*, and in *Managing Clusters of HP 9000 Computers Sharing the HP-UX File System*. You should read them and in that order.

Problem Areas

Problems with HP-UX cluster operation can occur in the following areas:

- System Boot-up (cluster server)
- System Boot-up (clients)
- System Panics
- LAN Problems
- CDF Mix-ups
- Configuration/Clusterization Problems

System Boot-up Problems

For system boot-up problems that can apply to all systems (standalone, cluster servers and cluster clients), see Chapter 5, “System Boot-Up Problems”.

Troubleshooting problems with clients that won't boot

HP-UX Cluster servers are disk-based systems and act very much like standalone systems during the boot up process. HP-UX clients, on the other hand, do not have an attached disk from which to get their operating system. They must receive their kernel over LAN from a cluster server. The daemon `rbootd` runs on the cluster server and handles boot requests from HP-UX client nodes. This adds a little complexity to the boot up process and a few more areas where things can go wrong.

If you are having problems with booting an HP-UX client node, check the following things:

- The client node is listed in the server's `/etc/clusterconf` file.
- The `/etc/clusterconf` file has the correct syntax (you check this using the command `ccck(1m)`).
- The kernel parameters associated with HP-UX clusters are set up properly.
 - *For information on how the kernel parameters should be set:* see Appendix A, "System Parameters" in the *System Administration Tasks* manual (see the section of the appendix called "**Cluster Related Parameters**)."
 - *To view/modify how your cluster related kernel parameters are currently set:*
 1. run `sam`
 2. Highlight **Kernel Configuration** and activate the **Open** control button.
 3. Highlight **Configurable Parameters** and activate the **Open** control button. A list of configurable kernel parameters will be displayed.
 4. From the "View" menu (on the menu bar), choose **Filter**. A "Filter" panel will be displayed.
 5. Set the "Operator" field for the item "Class" to contain "Matches".
 6. Set the "Value" field for the item "Class" to "Cluster" (be sure to observe the capital letter C in the word cluster).
 7. Activate the **OK** button. You should now have only the cluster-related system parameters displayed.
 8. You can now view how you have the cluster-related parameters for your kernel set.
 9. (Optional) You can change the value of a parameter by highlighting its entry from the displayed list, and choosing **Modify Configurable Parameter...** from the "Actions" menu (on the menu bar).

Local Disk Boot-up Problem

If a disk (local to a client computer) has a boot area on it, the client computer may try to boot from the local disk, instead of the cluster server. For details on how to handle this, see “System Boots From Local Disk Instead of the Cluster Server” in Chapter 5 in Chapter 5, “System Boot-Up Problems”.

System Panics

System panics are covered in Chapter 10, “System Panics” of this manual. There are several conditions specific to HP-UX clusters that can cause System Panics.

Condition	Reason
Lost contact with server	A client node has lost contact with its server. This is usually the result of LAN problems such as a disconnected cable or the server has gone down. A <i>server</i> will not panic because it has lost contact with its clients.
NFS incompatibility	If one node in a cluster has NFS configured in its kernel, then ALL nodes in the cluster must have NFS configured. If the server has NFS configured and a client you are trying to boot does not, the client node will panic on boot-up. Adjust the NFS status of all the kernels used in your cluster so that <i>all</i> or <i>none</i> of the kernels running in the cluster have NFS.
CD-ROM incompatibility	If one node in a cluster has CD-ROM configured in its kernel, then ALL nodes in the cluster must have CD-ROM configured. If the server has CD-ROM configured and a client you are trying to boot does not, the client node will panic on boot-up. Adjust the CD-ROM status of all the kernels used in your cluster so that <i>all</i> or <i>none</i> of the kernels in the cluster have CD-ROM.
Incorrectly configured swap space	If the server has the client node configured for local swap (local to the client node) but the client node does not have a disk, the client will panic on boot-up.

LAN Problems

HP-UX clusters are implemented using a low-level protocol to pass information/messages over LAN between the various nodes in the cluster. Because HP-UX clusters are so heavily dependent on the LAN, they are vulnerable to many of the problems that can occur in LAN configurations. Problems such as those listed here can cause HP-UX clusters not to function:

- Broken LAN cable
- Improperly terminated LAN cable (each end of the LAN *must* have a 50 ohm terminator or the LAN will not function properly, if at all)
- Extremely heavy LAN traffic
- Bad LAN connections/hardware
- Improper LAN configurations

CDF Mix-ups

If you use **SAM** to configure your cluster, you shouldn't run into problems in this area too often. If you manually create your own CDFs (for new programs, etc.), you might accidentally place the *contents* of a file in the wrong *context* of the CDF. For example, in a mixed cluster (consisting of Series 300/400 and Series 700 computers), you might have created a program (on one of the clients) that you want to make available to all of the Series 300/400 clients in the cluster. The *context* that you should use is the processor type. But, if you simply copy the executable to the CDF (as in the example below), **autocreation** will make the file `/users/bin/proga+/yournodename`. For information on autocreation, see Chapter 2, "Understanding Clusters" in the manual *Managing Clusters of HP 9000 Computers*.

```
cp a.out /users/bin/proga      Note: /users/bin/proga is a CDF
```

This program will then be accessible *only* to the system `yournodename` and *not* to the other Series 300s/400s in the cluster. The proper command to use is:

```
cp a.out /users/bin/proga+/HP-MC68020
```

If, due to a CDF mix-up, you attempt to execute a command that doesn't match the architecture of the system (for example, a command is compiled on

a Series 700 computer and you try to use it on a Series 300/400 computer), you will see the error message "Executable file incompatible with hardware." Some useful tools to help you locate/correct problems with CDFs are:

Tool	Function
<i>find</i> (1)	The "-hidden" and "-type H" options allow you to locate files that are CDFs. The two options are <i>not</i> synonymous. The "-hidden" option causes <i>find</i> to include elements of hidden directories (CDFs) in its search. The "-type H" option causes <i>find</i> to match on files that are CDF hidden directories.
<i>file</i> (1)	The <i>file</i> command attempts to classify a file by examining its contents. It can usually identify (and display) which files are Series 300/400 files and which files are HP-PA (Series 700/800) files. <ul style="list-style-type: none">■ Series 300/400 program files will be listed as "s200 executable"■ Series 700/800 program files will be listed as "s800 executable"■ Shell script files are usually listed as "commands text"■ Other text files (such as <i>/etc/passwd</i>) are usually listed as "ascii text"
<i>ls</i> (1) and <i>ll</i> (1)	The "-H" option causes these commands to print a "+" after any file that is a CDF. The "+" is <i>not</i> part of the file name; it is an indicator that this file is a CDF. Actually, in the output of an <i>ll</i> command, the "s" (located in the permissions field, indicating that the SETUID bit is set <i>for a directory</i>) is the <i>true</i> indicator that this file is a context-dependent file. You will also notice an "H" in column 1 of these entries. The permissions field for the file in <i>ll</i> 's output will look something like <i>Hrwsr-xr-x ...</i> <p>If these commands are used on CDF's themselves (as opposed to directories containing CDF's), then the elements of the CDF are displayed (similar to <i>showcdf</i>).</p>

showcdf(1)

Based on the current contents of a hidden directory and the context string for the computer where the command was executed, **showcdf** will list the name of the file (the element) within the hidden directory that matches the context of your computer. This is very helpful in determining which file within the CDF is being matched (if any) by other commands. Here are two examples (the first shows which element of the CDF **/etc/inittab** is being used by the computer where **showcdf** was executed. The second shows which element of the CDF **/lib** is being used by the computer where **showcdf** was executed.):

```
showcdf /etc/inittab
```

```
/etc/inittab+/hpxyz
```

```
showcdf /lib
```

```
/lib+/HP-PA
```

makecdf(1m) Converts a “normal” file to a CDF and allows you to specify the context for the contents of the original file. This is the safest way to manually create a CDF if you must do so. For a description of how to use *makecdf*, refer to the *makecdf(1m)* manual reference page and to *Managing Clusters of HP 9000 Computers Sharing HP-UX File Systems*.

CDF confusion can also occur when you have a file that has a “+” sign as the last character in its file name. This is *legal* because (as previously mentioned in the discussion of the *ls* command) the *true* indicator that a file is a CDF is that:

1. It has its SETUID bit set.
2. It is a directory.

To avoid confusion, it is best not to use “+” signs in your file names, especially as the last character.

It is possible for it to appear that a file doesn’t exist. This happens when the file is a CDF and there isn’t an element of the CDF that matches the context of the computer that you are running on. Using *ls -H* ensures that a CDF is *always* shown.

Configuration/Clustering Problems

Problems with *init*

In setting up an HP-UX cluster, there are many details that must be attended to. Most of these are handled for you by the SAM utility. During configuration, it is possible to make a mistake in the data entry phase for LAN Link Level Addresses, IP addresses, and other information. If the information in the */etc/clusterconf* file (created by SAM during cluster configuration) is incorrect, the *init* program on the cluster server will complain at boot time. You might see a message similar to:

```
INIT: WARNING: LAN hardware inconsistent with /etc/clusterconf
```

The above error message indicates that the LINK LEVEL ADDRESS (LLA) listed in the file */etc/clusterconf* does not match that of your LAN interface card.

A system's **context** is set during boot time using the contents of the file `/etc/clusterconf`. If you remove or corrupt `/etc/clusterconf` from a running system, there will be no effect on the system's context until the next time that system is booted. You can verify that the entries in `/etc/clusterconf` have the correct syntax by using the command `ccck(1m)`. For detailed information on what the contents of `/etc/clusterconf` should look like, see Chapter 8, "Introduction to Cluster Administration" in the manual *Managing Clusters of HP 9000 Computers*.

Note Note the exact wording of the error message, and based on its text, correct the problem as soon as possible. If you do not, error messages such as "invalid argument" might appear at clusterization time.

"Failed kernel selftest" errors

- Failed kernel selftest: Cannot allocate file system buffer.
- Failed kernel selftest: Cannot allocate kernel network buffer.
- Failed kernel selftest: Cannot allocate kernel message buffer.
- Failed kernel selftest: Cannot invoke CSP.

These error conditions are probably caused by incorrect configuration of diskless kernel parameters, but they could also indicate that something is seriously wrong (such as a hardware failure). Check the log file `/usr/adm/errlog` for possible configuration problems. *The directory /usr/adm is a CDF!* Be sure to check the element of the CDF matching the context of your computer.

Note After using SAM to configure the kernel on a client, be sure to reboot the client before configuring the kernel on another client. If you fail to reboot the first client, the kernel that you made will be overwritten by the second client kernel information, and the first client kernel will not be installed.

HP-UX Cluster Log Files

The following files are used by the HP-UX cluster software to log information about its activities and status. They are useful troubleshooting tools.

file	used for
<code>/usr/adm/errorlog</code>	Contains information regarding possible configuration problems. <i>The directory /usr/adm is a CDF!</i> Be sure you examine the element of this CDF that matches the context of your computer.
<code>/usr/adm/rbootd.log</code>	Contains information logged by the <code>rbootd</code> daemon. See the manual reference page for <code>rbootd(1m)</code> ; it contains information on setting <code>rbootd</code> 's logging level.
<code>/tmp/cluster.log</code>	Used by the <code>SAM</code> utility to record its actions and any associated errors while creating a cluster (or when adding and removing <code>cnodes</code>).

Allowing Users Shutdown Capabilities

A user does not need to be superuser to halt or reboot a cluster node. You can use the `/etc/shutdown.allow` file to give permission for specific users to shut down specific computers in the cluster.

What you will probably want to do is to allow the “owners” of workstations in the cluster to shut down their own local nodes. You give this permission by entering the user login name and the cluster node name in `/etc/shutdown.allow`. For example, to allow user `fred` to shut down `client1`, make the following entry in `/etc/shutdown.allow`:

```
client1 fred
```

The superuser, and possibly other users, will need to be able to shut down all the cluster nodes, and, on the other hand, you might want to allow some cluster nodes to be shut down by anyone. You can use a wildcard character in such cases. For example, to allow the superuser to shut down all the cluster nodes, make the following entry in `/etc/shutdown.allow`:

+ root

The entry for *shutdown(1M)* in the *HP-UX Reference Manual* contains more information.

Caution

Be careful when adding entries to `/etc/shutdown.allow`. This is how it works:

- If `/etc/shutdown.allow` does not exist, or exists and is empty (the default), then the superuser, and only the superuser, can shut down any cluster node.
- If `/etc/shutdown.allow` is not empty, then only those users listed can execute the `shutdown` command, and they can shut down only those systems listed beside their login names.

If you use `/etc/shutdown.allow`, you must make sure that it contains all the permissions you want or need to grant, *including the superuser login and the systems the superuser can shut down.*

In the worst case, an `/etc/shutdown.allow` file containing only a garbage entry would prevent anyone from shutting down anything.

- An entry in `/etc/shutdown.allow` allows a user to halt or reboot the system named, but not to bring it down to single-user mode. This capability is reserved for the superuser.
-

Where to go for more information

- For information on HP-UX cluster concepts, refer to *How HP-UX Works: Concepts for the System Administrator*.
- For detailed information on setting up and administering an HP-UX cluster refer to *Managing Clusters of HP 9000 Computers Sharing the HP-UX File System*.
- For information on setting up a LAN and other networking information, see the following manuals:
 - *Networking Overview*
 - *Installing and Administering LAN/9000*
 - *Installing and Administering FDDI/9000 Software*
 - *Installing and Administering Token Ring/9000 Software*
 - *Installing and Administering NFS Services*
 - *Installing and Administering Network Services*
 - *Installing and Administering ARPA Services*

System Boot-up Problems

What Happens During the System Boot-up Process

Booting your system refers to the process of getting your computer up and running (from a previously halted state) with the operating system (in this case HP-UX) in control of the computer. The boot process for any HP-UX system can be summarized in four phases:

1. Boot ROM initializes/tests hardware
2. Boot ROM loads (and runs) a small secondary loader program
3. The secondary loader program loads (and runs) an HP-UX kernel
4. HP-UX begins running

Each of the phases in the boot process has its own set of problems associated with it. A good method for troubleshooting system boot-up problems is to try to identify the phase of the boot process where the boot failed and then look at what things can go wrong during that phase. You could also do it the other way around, by looking at the types of things that can go wrong in each phase and then try to match those things to your specific situation.

The remainder of this chapter will discuss each of the boot phases, what symptoms you might see if a problem occurred during that phase, and what you can do to fix the problem.

Quick Reference Table (Start Here)

To help you locate your problem as quickly as possible, use this table to locate the symptom that you are seeing. The table will provide you with a short list of things to check and a reference to the point in this chapter to go for more detailed information.

Table 5-1. System Boot-up Problems (Quick Reference)

Series	Symptom	Things to Check or Try	For Details Refer to:
All	No Lights on front panel, no fan noise, no disk noise, no other signs of activity	<ul style="list-style-type: none">■ Check power connections to computer, monitor■ Check power switches (on?)■ Check fuses on all equipment■ Possible hardware problem	Phase 1
All	Computer front panel lights are on, but nothing appears on the console display	<ul style="list-style-type: none">■ Check power connections to console■ Check interface cable between computer and console■ If console is a terminal:<ul style="list-style-type: none">□ Incorrect cable between computer and console terminal□ Is the terminal in "Remote Mode"?□ Are data communication parameters set correctly?	Phase 1

**Table 5-1.
System Boot-up Problems (Quick Reference) (continued)**

Series	Symptom	Things to Check or Try	For Details Refer to:
All	A few lines of information appear on the console screen, but no further activity after that	<ol style="list-style-type: none"> 1. Record Information from the screen 2. Try resetting the computer 	Phase 1
Series 300/400	“Searching for a System” (No further activity)	<ul style="list-style-type: none"> ■ Is boot device powered up? ■ Is boot device bus address set correctly? 	Phase 2
Series 700	List of potential boot devices is empty	<ul style="list-style-type: none"> ■ Are cables to boot device connected? ■ If booting from LAN: <ul style="list-style-type: none"> □ Are computers correctly connected to LAN? □ Is the server computer running? □ Is the server computer running rbootd? □ Is the client computer configured on the server? ■ Possible corrupted boot area on disk. ■ SERIES 700 ONLY: <ul style="list-style-type: none"> □ Select “s” from menu □ If entries now listed, use “b” to boot from them. 	

**Table 5-1.
System Boot-up Problems (Quick Reference) (continued)**

Series	Symptom	Things to Check or Try	For Details Refer to:
Series 700	List of potential boot devices not empty, but no activity after list is displayed	<ul style="list-style-type: none"> ■ Are autoboot and autosearch turned off? Turn them on. ■ Use "b" menu option to continue boot 	Phase 1
Series 700	4 rows of hexadecimal numbers are displayed, I/O status message	Boot device is probably not responding: <ul style="list-style-type: none"> ■ Did you enter the correct hardware address? 	Phase 2
Series 800	FAILED to Initialize ENTRY_INIT STATUS message and 4 rows of hexadecimal numbers are displayed	<ul style="list-style-type: none"> ■ Is the boot device powered on? ■ Is the boot device properly connected? ■ Is the boot device in ready state (for example, media loaded)? ■ Is the boot device address set correctly? ■ Is the boot area corrupted on boot device? 	

5

**Table 5-1.
System Boot-up Problems (Quick Reference) (continued)**

Series	Symptom	Things to Check or Try	For Details Refer to:
ALL	/hp-ux: cannot open or not executable	<ul style="list-style-type: none"> ■ Try booting from a backup kernel (for example /SYSBCKUP) ■ Attach disk to working system (or boot from a recovery tape or good disk) □ Check for a valid kernel file (hp-ux) 	Phase 3
Series 700/800	iodc_open failure in fopen OR iodc_fopen: open failure	<ul style="list-style-type: none"> ■ Verify that you entered the correct address in the "hpux boot" command ■ If occurs during autoboot: <ul style="list-style-type: none"> □ Boot manually (specify correct device address) □ Fix auto execute file (LIF:AUTO) 	Phase 3

**Table 5-1.
System Boot-up Problems (Quick Reference) (continued)**

Series	Symptom	Things to Check or Try	For Details Refer to:
ALL	Miscellaneous error messages HP-UX had problems configuring hardware and/or binding drivers.	<p><i>If system boots:</i></p> <ol style="list-style-type: none"> 1. Log in 2. Review messages using the <code>/etc/dmesg</code> command 3. Check associated devices <p><i>If driver binding problems:</i></p> <ul style="list-style-type: none"> ■ Check for driver in kernel (check <code>dfile</code> or <code>S800</code> file). If drivers are missing, add them ■ Corrupted shared library? <p><i>If system won't boot:</i></p> <ul style="list-style-type: none"> ■ Check associated devices ■ Try to boot from an alternate kernel 	Phase 4
ALL	Kernel panics, message indicates primary swap space could not be configured	<ul style="list-style-type: none"> ■ Check device being used for primary swap <ul style="list-style-type: none"> □ Is device turned on? □ Is device connected? □ Device error condition? ■ Try booting from an alternate kernel (SYSBCKUP?) ■ SERIES 800 ONLY: <ul style="list-style-type: none"> □ Override the primary swap location with the <code>hpux -aS</code> option 	Phase 4

5

**Table 5-1.
System Boot-up Problems (Quick Reference) (continued)**

Series	Symptom	Things to Check or Try	For Details Refer to:
ALL	No login prompts on terminals	<ul style="list-style-type: none"> ■ Check entries for gettys in the <code>/etc/inittab</code> file ■ Verify run state with <code>/etc/who -r</code> command 	Phase 4
ALL	Miscellaneous error messages during processing of <code>/etc/rc</code> file	<ul style="list-style-type: none"> ■ Resent changes to <code>/etc/rc</code> file? ■ Missing or incorrect entries in <code>/etc/checklist</code> file? ■ Missing files on system? 	Phase 4
ALL	System booted, but doesn't look normal?	<ul style="list-style-type: none"> ■ Booted from correct kernel? ■ Booted from correct device? 	<p align="center">“This Doesn't Look Like My System (Strange Behavior After Boot)”</p>

Phase 1: Boot ROM Initializes Hardware

In this phase, the boot ROM tests and initializes the hardware (processors, coprocessors, internal buses, memory, etc.).

Note On PA-RISC computers, the boot ROM is part of ROM called “Processor Dependent Code” (PDC). On a Series 300 or a Series 400 computer, it is simply called the **boot ROM**.

What Can Go Wrong During Phase 1

Problems during this first phase of the boot-up process are rare. During this phase problems can be caused by:

5

- No power to the computer
- Processor hardware failure
- Interface card hardware failure
- Device failure

Symptoms (Phase 1 Problems)

Here are some possible symptoms of Phase 1 boot-up problems:

Symptom: *No lights, disk drive noises, fan noises, or other signs of activity.*

- Remedy:
1. Check your building’s circuit breakers and the power connections to your computer equipment to be sure that power is reaching your computer.
 2. Be sure that the power switches to *all* of your equipment (the computer, external disk drives, the monitor or console terminal, etc.) have been turned on.
 3. Check any customer-accessible fuses on your equipment. If they are found to be blown, replace them *only* with new fuses of equivalent value. If the new fuses immediately blow, have the defective device repaired by a qualified service person.

Symptom: *Lights on front of computer are on, but nothing appears on the console screen.*

Remedy: 1. Verify that your computer's monitor (or console terminal) is turned on and properly connected to your computer. If your console is a terminal, be sure that it is in "remote mode" and that the datacom parameters of the terminal are correctly set. In nearly all cases, the correct settings will be:

Baud Rate:	9600
Parity/Databits:	None/8
Chk Parity:	NO
EnqAck:	YES
CS (CB)Xmit:	NO
RecvPace:	Xon/Xoff
XmitPace:	Xon/Xoff

Configuration parameters that are not mentioned can be set to any value and do not affect the operation of the Console/LAN card or the system.

2. Wait. On some computers with large amounts of memory and lots of hardware to test, it takes a while to initialize all of the hardware. On some of the largest configurations, this could take up to 5 or 10 minutes. If your system has a chassis-code display (usually a 7-segment, 4-digit display) on it, observe whether or not the display is changing (or other lights are flashing on the front of the computer), indicating activity.
3. If after several minutes of no observable activity, or after 5 or 10 minutes of no console display, try resetting the computer. See the owners guide or operator's guide that came with your computer for the proper "reset" procedure. Generally, you can do this by pressing the "Reset" button (which might be labeled "TOC" on some computers), or by

turning off the power to your computer, waiting five or ten seconds and then turning the power back on.

4. If the problem reoccurs, record the symptoms, the status of any indicators (especially any LED displays) on your processor, and any messages that appear on your system console. You will need this information when you place a service call.

Symptom: *A few lines of information appear on the screen, but no apparent activity beyond that.* The information might or might not contain error messages

- Remedy:
1. Your computer might already be in phase two of the boot process, and unable to find a secondary loader program. See phase two for details on what you might expect to see when the boot program begins searching for a secondary loader program.
 2. If you don't think that your computer has started searching for a secondary loader program, record the information you see on your screen, particularly the last few lines (to indicate where in the process it stopped), and any error messages that are displayed.
 3. Try resetting the computer. See the owner's guide or operator's guide that came with your computer for the proper "reset" procedure. Generally, you can do this by pressing the "Reset" button (which might be labeled "TOC" on some computers), or by turning off the power to your computer, waiting five or ten seconds and then turning the power back on.
 4. If the problem reoccurs, be sure you have recorded the symptoms, the status of any indicators (especially any front-panel lights and LED displays) on your processor, and any messages that appear on your system console. You will need this information when you place a service call.

At this stage in the boot process, most of the problems that occur require your hardware to be serviced by a person trained and qualified to do so.

Phase 2: Boot ROM Loads the First Level of Software

In this phase the boot ROM attempts to load and run the first level of software. On a PA-RISC (Series 700/800) computer, this is known as the **Initial System Loader (ISL)**. On a Series 300 or Series 400 computer, this is known as the **secondary loader program**. Where the boot ROM looks for this software depends on what type of computer that the program is running. It is usually located on disk in a special area called a LIF (for *Logical Interchange Format*) volume, but can be located on tape or another device.

Note On PA-RISC computers, this phase actually consists of two parts. In the first part, the boot ROM loads and runs ISL (a system independent loader that does not know about the HP-UX operating system). ISL, in turn, loads a system-specific loader utility called **hpux**. Do not confuse “hpux” (the system-specific loader utility), with “/hp-ux” (the file that usually contains the HP-UX kernel), or with “**HP-UX**” (a generic reference to the HP-UX operating system).

- On a Series 300/400, the LIF volume is at the beginning of the disk.
- On a Series 700 computer, the LIF volume resides at the end of the disk; there are pointers to it located in a header area at the beginning of the disk.
- On a Series 800 computer, it resides in a special disk section often referred to as the “boot” partition. If your root file system is a logical volume, then ISL resides in a special “LIF Volume Area” near the beginning of the disk.

What Can Go Wrong During Phase 2

There are basically two things that can go wrong in this phase of the boot process:

1. The boot program cannot find the secondary loader program.
2. The boot program finds and loads the wrong secondary loader program.

Symptoms (Phase 2 Problems)

Symptoms of problems at this phase of the boot-up process vary greatly depending on the type of computer you have.

Series 300 and Series 400 Computers

The boot program has a specific order in which it searches for potential boot devices (devices containing a secondary loader program). The specific order can be found in Chapter 2, "System Startup", in the manual *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029. In general the boot program looks at devices in this order:

1. Disk drives (HP-IB or SCSI, with the highest priority it can find)
2. Shared Resource Manager (if present, at select code 21)
3. Local Area Networks (if present, at select code 21)
4. Memory (Bubble, EPROM, ROM, etc.)
5. Other disk drives

Because there is a particular sequence that the boot program uses to locate a secondary loader program, it may find one before it reaches the one you want it to actually use. The boot program will use the first one it finds.

Symptom: *Message: "SEARCHING FOR A SYSTEM (Return to pause)" is displayed. No further activity is apparent.*

When the boot program begins searching for the secondary loader program, it displays the message "SEARCHING FOR A SYSTEM (Return to pause)" on your monitor's screen. If the boot program can't locate a secondary loader, the message will remain on the screen. The computer will not appear to be doing anything, and the screen display will not change.

If the boot program finds the wrong secondary loader, the computer will probably attempt to load HP-UX from the wrong device or location. If it is not successful, the symptoms will be similar to those described in the previous paragraph. If it *is* successful at running the wrong secondary loader, that

secondary loader program will probably load the wrong kernel. This could lead to strange system behavior. For information about that behavior and what to do about it, see the section “This Doesn’t Look Like My System (Strange Behavior After Boot)”, at the end of this chapter.

Remedy:

1. Verify that your disk drives (or other devices that you want to boot from) are powered up, set to the correct bus address, and that the cables attaching them are firmly in place. See the manuals that came with your disk drives to see how to properly set their bus addresses, etc.
2. If you’re booting from another computer (over the LAN), be sure that:
 - a. The LAN cable is correctly connected to your computer. A properly connected LAN cable is terminated at both ends with a 50 ohm terminator. Computers should never be attached directly to the *ends* of a LAN cable; rather they should be attached via T-connectors or taps along the cable’s length.
 - b. The system from which you are booting is running, and is attached to your LAN.
 - c. The remote boot daemon (`/etc/rbootd`) is running on the system that you are attempting to boot from.

5

3. If you are attempting to boot from a local device (such as a disk drive); and, if the device is on, ready, and configured properly, there is a chance that your kernel file is missing or corrupted. There are several things you can do at this point:

a. If you have another system you that can attach the disk drive to, you can mount the file system to that system and check for the presence of a valid kernel.

b. If you have previously made a recovery system (using mkr), this is the very reason you made it. To use the recovery system:

i. Be sure that your tape drive (probably a cartridge tape drive or DDS format tape drive) is turned on.

ii. Put your recovery tape in your tape drive, wait for the tape drive to finish its search for the beginning of tape, and reset your computer according to the procedure in the Owner's Guide or Operator's Guide that came with your system (normally this is done by pressing the "Reset" button or "TOC" button on your computer, or by turning it off, waiting 5 or 10 seconds, and turning it back on).

Your system should now boot from the recovery tape.

iii. Once your system has booted from tape, you can mount the system disk and check for valid kernels in the root directory (which is now the directory where you mounted the disk to).

The purpose of a recovery system is to allow you to check for problems and repair them.

iv. Once you have made the necessary repairs to your system disk, you can (and should) reboot your system from the restored system disk.

Note

If you cannot find a recovery system or a bootable disk, *and* if you don't have another system that you can mount your disk on, you will need to re-install HP-UX following the same procedures that you did when you first got your computer. You will then need to restore your files from a recent full backup (to restore all of your customized files). When doing the restore, be sure to tell the restoration program that it should overwrite newer files with the older ones from the tape.

Series 700 Computers

Symptom: *The list of "Potential Boot Devices" is empty. You see a display that looks similar to this:*

```
Searching for Potential Boot Devices.  
To terminate search, press and hold the ESCAPE key.
```

```
Device Selection      Device Path          Device Type  
-----
```

```
b)  Boot from specified device  
s)  Search for bootable devices  
a)  Enter Boot Administration mode  
x)  Exit and continue boot sequence  
?)  Help
```

```
Select from menu:
```

Remedy: The empty list of potential boot devices indicates that the boot program could not locate any potential boot devices.

1. Select "s" from the menu. The empty list might be the result of your inadvertently hitting the **ESC** key during the initial search. Selecting "s" from the menu initiates a more thorough search for devices from which you can boot.

If the list is still empty, proceed to step 2.

If the new search located bootable devices, use the "b" option from the menu and the value from the "Device Selection" field in the list to boot from the desired device.

2. Verify that your disk drives (or other devices that you want to boot from) are powered up, set to the correct bus address, and that the cables attaching them are firmly in place.
3. If you're booting from another computer (over the LAN), be sure that:
 - a. The LAN cable is correctly connected to your computer. A properly connected LAN cable is terminated at both ends with a 50 ohm terminator. Computers should never be attached directly to the *ends* of a LAN cable; rather they should be attached via T-connectors or taps along the cable's length.
 - b. The system from which you are booting is running, attached to your LAN, and is configured to serve your system.
 - c. The remote boot daemon (`/etc/rbootd`) is running on the system that you are attempting to boot from.
4. If you are attempting to boot from a disk drive, the boot area (where the secondary loader program resides) might have been corrupted or the disk might not have one. See "Creating/Recreating a Boot Area on a Disk or Logical Volume" at the end of this chapter for information on how to create (or recreate) a boot area.

Symptom: *There ARE entries in the list, and you are presented with the above menu and prompt*

Remedy: You might have the **autoboot** and **autosearch** flags turned off. If autoboot and autosearch are turned off (or you have hit the **ESC** key), the boot program will display a list of possible boot devices and then present you with the menu of possible actions.

Use the “b” option to continue the boot process. If you would like your computer not to stop here in the future, enter boot administration mode and turn on the “autoboot” and “autosearch” flags.

If autoboot and autosearch flags are turned on (and the **ESC** key is not used to override the boot sequence), the boot program will attempt to locate ISL at the address defined as the primary boot path.

Symptom: *If you see a display similar to this:*

```
0000350B 00000012 FFFF1CF 80240000 0020000C 00000000 0021F000 00000400
9E030000 FFFFFFF0 0000E1AC 00000000 FFFFFFFF 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

I/O Status = -7

You probably attempted to boot your system from a device that does not respond or is not connected.

- Remedy:
1. Verify that the device you are trying to boot from is powered on and properly connected to your system.
 2. Verify that the device you are trying to boot from is in a “ready” state (for example if you are trying to boot from a DDS format drive, be sure that it has a tape in it that has ISL on it).
 3. If the device is a SCSI device (or HP-IB device), be sure that its bus address is set properly.

Symptom: “IPL Checksum Error” message is displayed.

Remedy: It is possible that the disk (or other device) that you are trying to boot from is responding, but does not contain a valid copy of ISL. See “Creating/Recreating a Boot Area on a Disk or Logical Volume”, later in this chapter for information on how to create/recreate a boot area (that contains ISL and its system-specific loader utility, hpux).

Series 800 Computers

If autoboot is enabled (and the ten second override is *not* activated), the boot program will attempt to locate ISL at the address defined as the primary boot path.

5

Otherwise, the boot program will ask the operator where to boot from. The primary boot path and the alternate boot path are displayed and offered as choices. If the operator answers “no” to both of these choices, the boot program will request a specific address from the operator and attempt to find ISL there.

The **primary boot path** and **alternate boot path** are hardware addresses that can be defined in stable storage (see the manual reference page for *isl(1m)* for information on how to display and modify these boot paths).

Symptom: *You see a console display that looks similar to this:*

```
Failed to Initialize
ENTRY_INIT status=-4

0B300041 00000002 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000381 0000BF02 00000000 00000000 00000070 00000002 0000003A 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Remedy: If the device that you told ISL to boot from does not respond, you will see display output similar to the above.

1. If you manually entered a hardware path for the system to boot from, be sure you entered the correct path for the device.
2. Check the status of the device where ISL resides:
 - a. Is it powered on and in a ready state (for example, if it is a removable disk drive, is it spun up and online?)
 - b. Is the device configured for the correct address (such as a SCSI or HP-IB address)?
 - c. Is the cable connecting it to the computer tightly connected to the correct location, and is it the correct cable?

Symptom: "IPL Checksum Error" message is displayed.

Remedy: It is possible that the disk (or other device) that you are trying to boot from is responding, but does not contain a valid copy of ISL. See "Creating/Recreating a Boot Area on a Disk or Logical Volume", later in this chapter for information on how to create/recreate a boot area (that contains ISL and its system-specific loader utility, *hpux*).

Phase 3: HP-UX is Loaded and Launched

In this phase, the secondary loader program (Series 300/400), or hpux (Series 700/800), attempts to load and start up HP-UX itself. As with the previous phase, there are two basic things that can go wrong:

1. The loader program can't find HP-UX
2. The loader program loads the wrong version of HP-UX

Symptoms (Phase 3 Problems - Can't Access Kernel)

As with Phase 2, symptoms at this phase vary depending on the type of computer you have.

5 Series 300 and Series 400 Computers

Symptom: *Message:* /hp-ux: cannot open, or not executable

Remedy: This indicates that the file containing the primary copy of the kernel (/hp-ux) is not present in the root file system or has been corrupted.

In order to fix this problem, you will need to boot from an alternate kernel. This requires an attended boot (turn the computer off and then back on while holding down the space bar until you see the word "keyboard" appear on your screen). A list of possible kernels will appear in the upper-right corner of your display. Try selecting the entry labeled "SYSBCKUP". This will attempt to boot your system from the file in your root directory of the same name (/SYSBCKUP). If no entries are present in the upper-right corner of your display (or if none of the listed entries work), see the next symptom for what to do.

Series 700 and Series 800 Computers

At this point, hpux is running and trying to locate the kernel that it is to load. A LIF file, called "AUTO" (also known as an auto-execute file), is located in the same LIF volume as hpux. The auto-execute file contains a string of characters that ISL uses to start up the hpux utility. This string of characters contains the name of the file that hpux is to use (from which it will load the

HP-UX kernel). This file is most often called “/hp-ux” and is usually located in a file system on the disk where hpux originated.

If you do not override the autoboot sequence (by pressing the **ESC** key on a Series 700 computer, or by pressing a key within 10 seconds on a Series 800 computer) ISL will read the auto-execute file and have hpux try to load HP-UX based on the information it finds there.

Whether hpux got its boot information from the auto-exec file, or you manually enter it at the “ISL>” prompt, several problems can occur:

Symptom: *A message is displayed similar to the following:*

```
iodc_open failure in fopen
```

OR

```
iodc_fopen: open failure
```

5

Remedy: This message indicates that hpux could not access the device from which it was told to retrieve the kernel.

- If you interactively entered the “hpux boot” command (from the “ISL>” prompt), verify that you entered the device address correctly.
- If the address in the auto-execute file is not correct, you can request to interact with ISL and manually boot from the correct location. You can replace the contents of your auto-execute file so that it reflects the correct boot location once your system is up and running. You can do this by using the “-a” option to the /etc/mkboot command, giving a string of characters that you want to be the new contents of the auto-execute file.

EXAMPLE (Series 700):

```
/etc/mkboot -a "hpux boot disk(scsi.5;0)/hp-ux" /dev/rdisk/c1d0s2
```

EXAMPLE (Series 800):

```
/etc/mkboot -a "hpux boot disk1(8.0.1;0)/hp-ux" /dev/rdisk/c1d0s2
```

Symptom: *A message such as `"/hp-ux: cannot open, or not executable"` is displayed.*

Remedy: This message indicates that hpux could not locate or open the file `/hp-ux`.

If you are trying to boot from a local disk (for example a disk at address `scsi.6.0` on a Series 700 computer), try listing the files in the root directory of the disk using the ISL command `"hpux ls"`.

EXAMPLE (Series 700):

```
ISL> hpux ls disk(scsi.6;0)/.
```

EXAMPLE (Series 800):

```
ISL> hpux ls disk1(8.0.1;0)/.
```

The hardware address (if you specify one), should be the one that you're trying to boot from. This might be different than those in the examples above.

You should see a listing of the files in the root directory of the device that you specify. Files in that directory having the executable permission flag set will have an asterisk ("`*`") appended to their name. Some of these are probably bootable kernels. Look for the name `"hp-ux"`. If it's not present, that may be the problem. Look for other possible kernels from which to boot (for example `SYSBCKUP`). Kernel files, in addition to having an asterisk behind their name, are also rather large in size. For Series 300/400 computers, kernel files will probably be between one and two megabytes in size. For PA-RISC computers, the kernel files will probably be larger than two megabytes.

For information about how to boot from a backup (an alternate) kernel, see the next section, "Symptoms (Phase 3 Problems - Finds Wrong Kernel)".

Symptoms (Phase 3 Problems - Finds Wrong Kernel)

For a variety of reasons, it is possible that the secondary loader program will load and run the wrong kernel file. If this happens, here's what you can do to override the selection of a kernel to boot.

Series 300/400 Computers

To temporarily boot from a different kernel, boot the system in attended mode. See Chapter 2, "System Startup" in the *System Administration Tasks* manual for details on how to do this.

The secondary loader program looks everywhere it can for valid kernel files from which to boot HP-UX. It uses the sequence listed earlier in this chapter (in the section called "Boot Program on a Series 300 Computer"). As it locates various kernel files, it creates a list of them and displays them on the screen in the order in which it found them. If you are booting in attended mode (held space bar down when you powered on the machine), this list remains on the screen and you have a chance to select from the list. If you are booting in autoboot mode, the system will immediately select the first kernel in the list. To be sure your computer automatically selects the kernel you want, you will need to make sure that your kernel of choice is the first one the secondary loader program will encounter.

Series 700 Computers

On a Series 700 Computer, you can override the selection of a boot device by performing the following steps:

1. Override the autosearch selection by hitting the **ESC** key on your keyboard when the computer prompts you to do so.
2. Override the autoboot selection by hitting the **ESC** a second time when the computer prompts you to do so.
3. Select "s" from the menu to initiate a formal search for all bootable devices
4. Select the "b" option from the menu to boot from the desired device

Note The above procedure will allow you to do a one-time boot from an alternate device. If you want to permanently change which device the computer boots from, enter boot administration mode (select “a” from the menu) and change the primary boot path.

Booting from an alternate Kernel on a Series 700 Computer. You can override the name of the kernel to use (default is hp-ux) by:

5. Performing steps one through three, as described above.
6. Specifying that you want to interact with ISL flag when you select the device to boot from
7. Entering the “hpux boot” command with the appropriate kernel name specified.

5

EXAMPLE:

```
Searching for Potential Boot Devices.  
To terminate search, press and hold the ESCAPE key.
```

Device Selection	Device Path	Device Type
-----	-----	-----
P0	scsi.6.0	C2472S

- b) Boot from specified device
- s) Search for bootable devices
- a) Enter Boot Administration mode
- x) Exit and continue boot sequence
- ?) Help

```
Select from menu: b P0 ISL
```

```
          :                  :  
          :                  :  
ISL> hpux boot disk(scsi.6.;0)/SYSBCKUP
```

The above example shows how to override the default file name (/hp-ux) when booting a Series 700 computer. The kernel file “/SYSBCKUP” is used instead of “/hp-ux”.

Series 800 Computers

On a Series 800 computer, you can override the selection of a boot device by performing the following steps:

1. Override autoboot by hitting a key on the keyboard within the ten second grace period.
2. Answer “n” to “Boot from primary boot path?”
3. Answer “n” to “Boot from alternate boot path?”
4. Specify the hardware path to boot from.

Note The above procedure will allow you to do a one-time boot from an alternate device. If you want to permanently change which device the computer boots from, you will need to change the primary boot path using the ISL “primpath” command.

You can override the name of the kernel to use (default is hp-ux) by performing the following steps:

1. Override autoboot by hitting a key on the keyboard within the ten second grace period
2. Answer the “Boot path” questions appropriately.
3. Answer “y” to “Interact with ISL?”
4. At the ISL> prompt enter:

hpux {newkernelname}

where {newkernelname} is the device specification and the name of the file containing the kernel you want to boot. For details on how to do this, see “Selecting a System to Boot” in Chapter 3, “Starting and Stopping HP-UX” in the *System Administration Tasks* manual.

For more information on your Series 800 options for booting alternate kernels, refer to the *HPUXBOOT(1m)* manual reference page.

Phase 4: HP-UX Begins Running

A lot of things happen during this final phase of the boot process. HP-UX takes control of the system's resources and:

- Configures all of the hardware on the system
- Binds the various drivers to the hardware components it has found
- Configures its primary swap location
- Locates and mounts the root file system
- Starts its first process ("init")

What Can Go Wrong During Phase 4

5 Because so much is occurring in this phase, there are a number of things that can go wrong here. Fortunately, most of the the things that can go wrong in this phase will not prevent you from booting. They may, however, limit what you can do with your system when it is up and running.

- As HP-UX is configuring all of the hardware on the system, there might be certain pieces of hardware that it cannot initialize. You will probably see error messages printed on the system console if this occurs, but most of the hardware that is *required* to boot, has already been accounted for during the previous phases of the boot process.
- As HP-UX is attempting to bind drivers to hardware, you might see error messages indicating that it could not bind a particular driver.
- If HP-UX is unable to configure its primary swap space, and is unable to configure another area to swap to, the kernel will panic.

Symptoms (Phase 4 Problems)

Symptom: *Miscellaneous error messages displayed on system console, indicating problems with configuring hardware and/or binding drivers to hardware.*

Remedy: Most of the errors that are represented by the displayed error messages will not prevent you from booting.

If your system continues to boot up and run:

1. Log in to your system.
2. Use the `/etc/dmesg` command to redisplay boot messages.
3. Use the driver names and any other information that is displayed to determine which piece (or pieces) of hardware are associated with the problem.
4. Check the connections to the affected pieces of hardware, and be sure the hardware is in working order and powered on.
5. If the error messages indicate a problem binding the drivers, the problem might be that a driver is missing from your kernel. Use SAM, or check the kernel configuration file (usually `/etc/conf/dfile` or `/etc/conf/gen/S800`) that was used to create your kernel to verify that the drivers in question were included in the kernel. If they are missing, you will need to add them, regenerate your kernel, and reboot your system using the new kernel file.
6. You might have a problem with a corrupted shared library. If the error messages lead you to believe that this has happened, try restoring the shared libraries from a recent backup.

If your system does *not* continue to boot:

1. Use the displayed error messages to determine which piece (or pieces) of hardware are associated with the problem.
2. Check the connections to the associated hardware, and verify that it is powered on and working (no error indicators).
3. If the problem seems to be associated with driver binding, you might not have a needed driver generated into your kernel. You will need to boot your system from an alternate kernel file or device and generate a new kernel that includes the needed driver(s).

Symptom: *Kernel panics with an error indicating that primary swap space could not be configured.*

5

Remedy: Most likely, the problem is that the device that you configured your kernel to use as a primary swap device is not powered on, or connected to your system. Its hardware address might have been changed. Check to be sure that the disk drive is connected, powered on, and has no error indicators lit.

The kernel might not reference the correct (or any) device for primary swap. If you suspect this might be the problem, try one of the following things (listed in order of preference):

- Boot from a different kernel (for example, /SYSBCKUP).
- Use the “-aS” option with the “hpux boot” command to specify a specific (working) device for the primary swap area. See *hpux_800(1M)* in the *HP-UX Reference Manual* for more details on how to do this.

Symptom: *No login prompts on terminals.*

Remedy: When HP-UX starts up its first process: a process called “init”. “init” processes the file `/etc/inittab` to know which processes it should run based on which run level HP-UX is in. In particular, there are processes called “gettys” that are responsible for putting login prompts on terminals. If you do not see a login prompt at a terminal where you think there should be one, check to be sure `/etc/inittab` has an entry for the getty corresponding to the run level HP-UX is currently in. For more information on problems of this type, see Chapter 9, “Unresponsive Terminals” in this manual.

Note

If your computer is part of an HP-UX cluster and you need to edit the `/etc/inittab` file, remember that `/etc/inittab` is a context-dependent file (CDF). Problems could occur if you accidentally change the wrong element of the CDF.

Symptom: *Miscellaneous error messages during the remainder of the boot process (for example, error messages related to the configuration of subsystems such as networking).*

Remedy: As `init` processes `/etc/inittab`, it executes a number of shell programs using scripts such as `/etc/bcheckrc`, `/etc/brc`, and (most notably) `/etc/rc`. The functions of these shell scripts are described in Chapter 2 of *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029 called Chapter 2, “System Startup”.

Most of the startup processing is done within the `/etc/rc` script, which, in turn, calls other scripts.

At this point your system is actually already booted. Any problems your system encounters as it executes these scripts are the same types of problems found in a running system. For example, in the `/etc/bcheckrc` script, the `fsck` utility might encounter problems with a file system. If this is the case, refer to Chapter 6 of this manual (File System Problems) for further assistance.

This Doesn't Look Like My System (Strange Behavior After Boot)

Symptom: *System seems to have booted normally, but it behaves unusually after it is up and running.*

If the boot program *is* successful at loading and running the wrong copy of HP-UX, the system might appear to have booted normally but weird behavior might be observed. For example:

- You might not be able to log in (some or all of the passwords might *appear* to have changed)
- You can't use certain devices that you used to be able to use (it appears that certain drivers are missing from your kernel)
- The system panics (complaining about an incompatibility of subsystems)
- A login prompt, other than the normal login prompt, appears

These symptoms can be an indication that you have booted the wrong version of HP-UX.

If your computer is a client in an HP-UX cluster and is booting from a LAN where more than one valid cluster server exists, the FIRST server to respond is the one that your system will boot from. This might *not* be the one you want and, in this case, which server responds first is unpredictable and can vary from boot to boot. For a server to respond to a boot request from your computer, it must have your computer configured in its `/etc/clusterconf` file. Unless your configuration requires the redundancy, you should never have more than one server on your LAN with the system you are now trying to boot defined in its `/etc/clusterconf` file.

A similar problem can occur if your client has a disk with an HP-UX operating system on it. You might have intended to boot from the LAN but instead have booted from the local disk (or *vica versa*).

Yet another possibility is that the server is using the wrong context of the CDF, `/hp-ux`.

Remedy: In all of the above cases, the solution is to reboot your computer in attended mode (overriding the autoboot sequence), and manually specify which device/computer/kernel to boot from. If you are not familiar with how to do this, refer to Chapter 3, “Starting and Stopping HP-UX” in the *System Administration Tasks* manual. See the next section, “System Boots From Local Disk Instead of the Cluster Server”, for information about how to permanently change where your system automatically boots from.

System Boots From Local Disk Instead of the Cluster Server

In the case of an HP-UX cluster, if a disk (local to a client computer) has a boot area on it, the client computer may try to boot from the local disk, instead of the cluster server.

5

Series 300/400 Computers

This is especially a problem with Series 300 and Series 400 computers. These computers have a specific order in which they search for bootable devices. The first place the boot programs look for a bootable device is an HP-IB disk with an HP-IB address of 0, or a SCSI disk with a SCSI device address of 7. If an HP-IB or a SCSI device (with the appropriate address) is present on your system, *and* if that device contains a boot area, your system will try to boot from that device.

Your options are:

- Boot in attended mode (hold the `Space Bar` down until you see the word keyboard, when you first start up the computer).
- Use SAM to build the file system on the disk (SAM does not put the boot programs at the beginning of the disk), so that the disk will not be a bootable disk.
- Use the “-n” option to the newfs command, so that the disk will not be a bootable disk.

- Set the HP-IB address to something other than zero (or the SCSI device address to something other than 7). This will cause these devices to be searched AFTER the LAN is searched. This allows the cluster server computer to respond to the boot request *before* the local disks are searched. This is probably the best option to choose as it allows your system to boot from a local disk if the cluster server does not respond. If you do not want your system to boot from a local disk (if the server doesn't respond) you can choose one of the other options in this list.

Series 700 Computers

If you do not override the auto-boot sequence (by pressing the **ESC** key), your computer will attempt to boot from the device defined by its primary boot path. If the boot path is pointing to a local device (such as a local disk), and if that device has a bootable operating system on it, your computer will boot from that device instead of the cluster server. If you want to usually boot from the cluster server, use the boot administration mode of your client system's boot program to change the primary boot path to point to "lan.0.255.0".

EXAMPLE:

```
path primary lan.0.255.0
```

Note

```
lan.0.255.0
  ^   ^   ^
  |   |   |
  |   |   |__accept default read retries
  |   |
  |   |_____keep trying to boot so don't
  |           have to reboot each client
  |           manually after server shutdown
  |
  |_____ server = (boot over LAN)
```

Recommended Settings for Boot Parameters

Creating/Recreating a Boot Area on a Disk or Logical Volume

If a disk or logical volume is to be used as a boot device, it must contain a special area called a boot area. A **boot area** contains the programs and information necessary to locate, load and run the HP-UX kernel. The most important of these programs is the secondary loader program (known as ISL and hpux on Series 700 and Series 800 computers).

If a disk has never been used as a boot device, or if the boot area has been corrupted or destroyed, you cannot boot from that disk. To create/recreate the boot area, use the *mkboot(1M)* command.

Here is the procedure:

1. If you are trying to create/recreate the boot area on a disk that currently does not boot, do *one* of the following:
 - Attach a *bootable* disk to your system, boot from it, and attach your non-bootable disk to the working system.
 - If your system is a Series 300/400 or a Series 700, boot your system from a recovery system (made previously from your system, using the `mkr` command). Unfortunately, the `mkr` command is not available on Series 800 computers.
 - Attach your non-bootable disk to another (working) system that has the appropriate boot programs for your computer type (see note).

Note

The `mkboot` program uses the following files (they contain the programs and information that gets written to the boot area):

Table 5-2.
Files containing boot information for mkboot

To Create a Bootable Disk for:	File Name
Series 300/400	<code>/etc/boot</code>
Series 700	<code>/usr/lib/uxbootlf.700</code>
Series 800	<code>/usr/lib/uxbootlf</code>

The appropriate file for your computer type must be on the system that you will use to run the `mkboot` command.

EXAMPLE:

If you will be using `mkboot` on a Series 800 computer to restore the boot area of a disk that will be used on a Series 700 computer, you must be sure that the Series 800 computer has the file `/usr/lib/uxbootlf.700` so that `mkboot` can use its contents to create the Series 700 boot area on the disk for the Series 700 computer.

2. *Do not perform this step without first reading the cautions below.*

Install the boot programs on the device that you want to boot from using the following command:

```
/etc/mkboot -v device
```

where:

device is the full path name to the device file for the disk to which you are installing the boot programs.

Caution

- Be *very* careful to specify the correct device file name when using the `mkboot` command. If you specify a valid (but incorrect) device (such as that of a different disk), you might overwrite valuable data.
- On Series 700 systems, a file system must reside on the disk being modified, so that `mkboot` can determine the layout of the disk.
- On Series 700 systems, the boot area is taken from swap space. `mkboot` cannot increase the amount of space allocated to boot programs on a disk where swap and or raw I/O are currently *enabled*. If you need to do this, refer to the `mkboot(1M)` man page (see the “-h” option).
- If the disk to which you are installing the boot programs is (or will be) an LVM disk (physical volume), *device* must specify section two. If any of your root file system, primary swap area, or dump area will be on a logical volume, there are additional things you must do to prepare the disk to be your boot disk. For specific information about what to do, see the section called “Moving Root from a Disk Section to a Logical Volume.” The section is located in Chapter 8, “Managing Logical Volumes” of the *System Administration Tasks* manual.

EXAMPLES:

Here are some examples of typical mkboot commands (your devices might be different):

Series 300/400/700:

```
/etc/mkboot -v /dev/rdisk/0s0
```

Series 800 (Non-LVM disk):

```
/etc/mkboot -v /dev/rdisk/c0d0s6
```

Note: On Non-LVM disks, section six is used as the boot partition

Series 800 (LVM disk):

```
/etc/mkboot -v /dev/rdisk/c0d0s2
```

Note: On LVM disks (physical volumes) you must use section two, and the disk must have been previously initialized with pvcreate's -B option

5

Where to go for more information

- For help with system bootup problems when your root file system is on an LVM disk, see Chapter 8, “Logical Volume Manager (LVM) Problems” in this manual.
- Detailed procedures for starting up and shutting down your system can be found in Chapter 3, “Starting and Stopping HP-UX” in the *System Administration Tasks* manual.
- For more information on the system startup sequence, the `init` process, and the `/etc/rc` file, see Chapter 2, “System Startup” in *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029.

File System Problems

File System Overview

The HP-UX operating system is *heavily* dependent on a functional file system. If a file system becomes corrupted (especially if it is the root file system) your system might panic or corrupt even more of the data on your disk. For this reason, you should regularly check the integrity of your file system(s) and correct any problems you find.

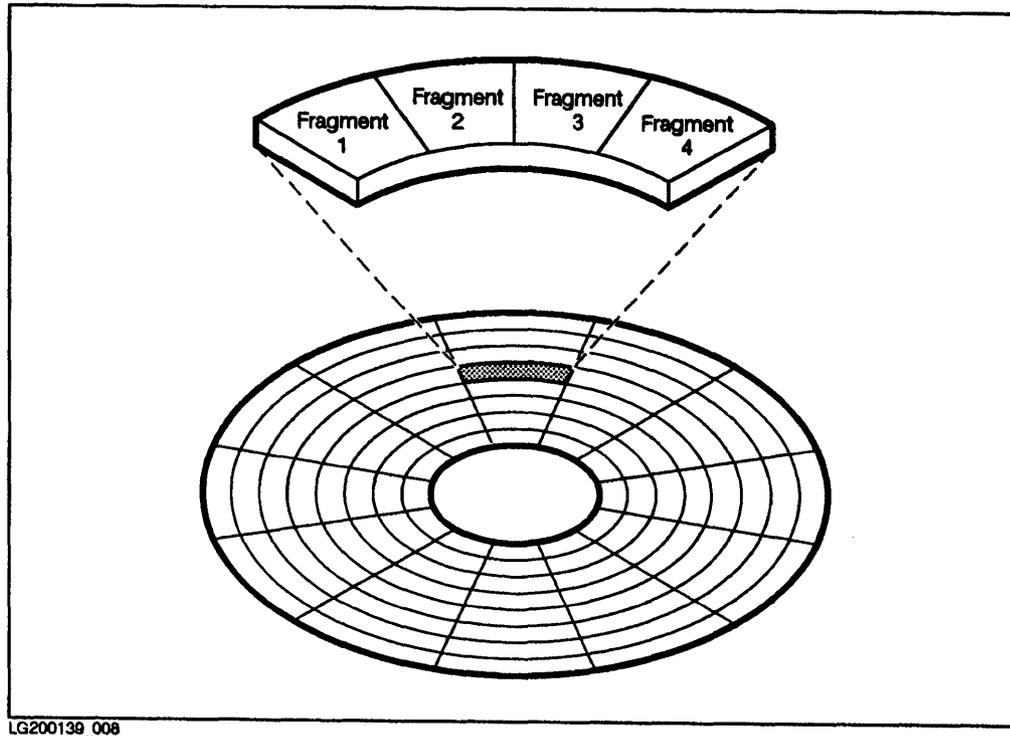
There are tools to help you check and correct file system problems, but to understand their usage and output, you need to know a bit about how file systems are organized.

File System Structure

The *logical* layout of a file system consists of the familiar directory tree containing directories and files. How this is *physically* implemented is the topic of this chapter.

A file system is constructed from a set of disk blocks. On an 800 series computer, this would be all of the disk blocks contained in a disk section (sometimes called a disk partition), or a logical volume. On a 300 series computer, this would be all of the disk blocks on a disk drive not associated with the boot or swap areas.

Each file system begins with a special disk block known as a **superblock**. The superblock contains global information about its file system, such as block and fragment size. A **fragment** is the smallest addressable part of a file system. Disk blocks are made up of fragments.



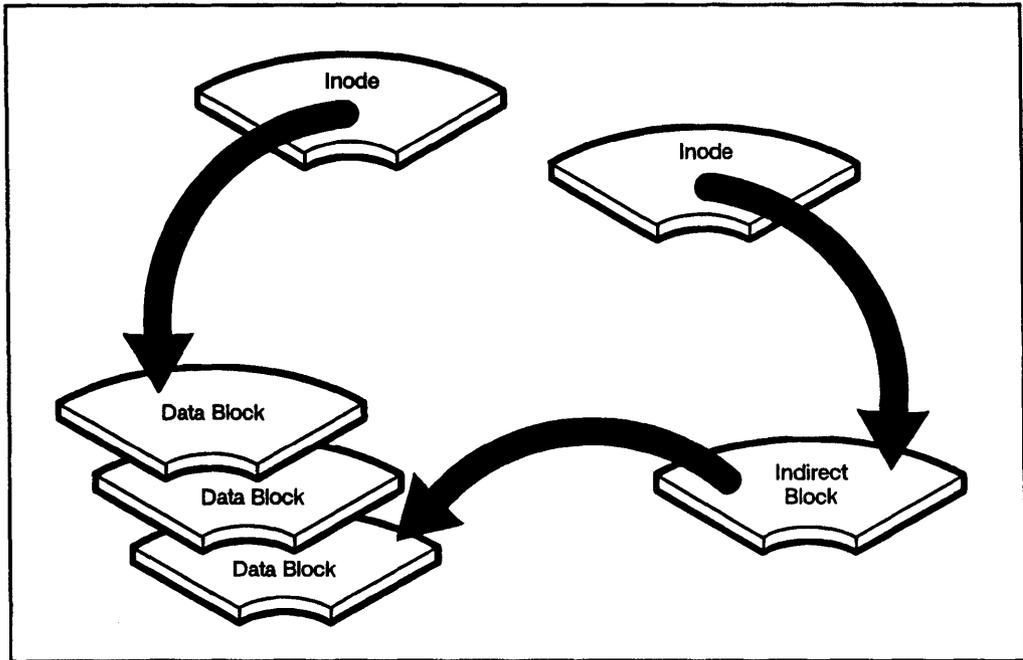
LG200139_008

Figure 6-1. Disk Block/Fragments

The superblock is in a known location (the primary copy of the superblock is the first disk block in a file system), so it is the starting block for the chains of disk blocks that make up the file system. Because of the importance of the information contained in the superblock, redundant copies of it are kept at specific (known) locations. If something happens to the primary superblock, one of these copies can be used to reconstruct the file system (assuming that the rest of the file system remained intact).

The disk blocks associated with a file system are used in several ways. In addition to the superblock, there are also inodes, data blocks, and indirect blocks. **Inodes** are disk blocks that contain information about the file/directory they are associated with: information such as the file's size, access permissions and ownership. Inodes also contain pointers to the data blocks associated with their file or directory.

6-2 File System Problems



LG200139_008

Figure 6-2. Disk Block Types

Data blocks are disk blocks that contain actual data for a file or directory. There are places in an inode for pointers to 12 data blocks. When a file is too big to fit in 12 data blocks, **indirect blocks** are used to point to additional data blocks.

File System Buffer Caching

For performance reasons, file system writes are cached (written to a memory buffer until there is enough data for an efficient write to physical disk or until the memory buffer is flushed to disk by a process called “syncer”). Until the data in memory are written to disk, there is an inconsistency between what HP-UX “thinks” the disk looks like and what it *actually* looks like. If the system is improperly shut down or crashes during this time, a file system can become corrupt. This is a common cause of file system corruption.

The Problem Areas

File system problems usually fall into one of these areas

- File system corruption
- Mounting/unmounting problems
- File system overflow (Refer to Chapter 7, “Disk Space Problems”)

Important If your computer is part of an HP-UX cluster, *always* diagnose and correct file system problems from the cluster server, *NEVER* from a client!

Defining the problem

If you suspect you have a file system problem, define it by answering the following questions:

- What command/application were you using when you noticed the problem?
- What didn't happen that should have?
- What did happen that shouldn't have?
- What files were you working with (specify their FULL path name)?
- Has anyone else experienced similar problems? With which files?
- What error messages were displayed (write down their exact wording)?

If the command you were using when you experienced the problem was `mount` or `umount`, skip to the section (later in this chapter) called “Problems with mounting and unmounting file systems”.

Identifying symptoms of a corrupt file system

If one of your file systems has been corrupted, you *might* notice some of the following symptoms. It is possible for a file system to be slightly corrupted without exhibiting symptoms.

Note

These symptoms do not always mean that one of your file systems is corrupt, but if you do notice one or more of them and can't identify another cause, check for file system integrity (see below).

In the following list, items in parentheses indicate examples of other possible causes of the symptom. You should check for these and other similar causes first. The HFS file system is robust and many times, the cause of your problem will be something other than a corrupt file system.

- A file contains incorrect or garbage data. (*It might have been overwritten.*)
- A file has been truncated or is missing data. (*A program error, someone using an editor or the truncate command might have caused this.*)
- Files disappear or change locations for unknown reasons. (*Someone might have deleted or moved the files.*)
- Error messages indicating file system corruption appear on a user's terminal or the system console.
- Has anyone else experienced similar problems?

Locating and correcting file system corruption

The primary tool for finding and correcting errors in HP-UX file systems is **fsck** (File System Checker). You can run **fsck** in several modes. The mode you select determines your level of interaction with **fsck** and its level of corrective action.

Caution

Some operations with **fsck** can cause loss of data. Read through the rest of this section *before* running **fsck**. Because of file system buffer caching (discussed earlier in this chapter) and because **fsck** does its work in several passes, you should always run **fsck** on a quiescent, unmounted file system. Running **fsck** on an active file system can corrupt it.

If, after considering and ruling out other possible causes for the problem, you suspect that you have a corrupt file system, perform the steps on the next few pages to locate and (if necessary) correct the problem.

As you study the steps, consider the following example:

Example:

An application that had been running well now reports that one of its data files is in the "wrong format" and the application cannot read it. Several users have also reported files containing "garbage" data. Another user reported a file missing. At the same time still another user reports that the line printer is rapidly ejecting paper and printing garbage randomly on each page.

Prerequisites (Record appropriate information)

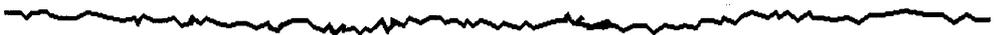
Remember to record the problem description for your records. It is best to treat each of the users problems separately in your records because it may turn out that their problems are unrelated. If they turn out to have a common cause, you can record that later. Entries in your log book might look something like this.

6

Date: _____ Time: _____
Who encountered the problem: _____
Problem Description:

Application "XYZ" reports an error
when trying operation "blah blah blah"
<exact wording of error message is ...>
The file it was trying to access was
/usr/XYZ/datafile.1

Resolution: _____



Date: _____ Time: _____
Who encountered the problem: _____
Problem Description:

Two users reported the following files have "garbage data" in them:

/usr/manufacture/ab.c
/usr/manufacture/de.f

A third user reported file /mnt/users/myfile was missing.

Resolution:____User #3's file /mnt/users/myfile was accidentally deleted by a co-worker, retrieved from backup tape.

Date:_____ Time:_____
Who encountered the problem: _____
Problem Description:

A fourth user reports that line printer "lp2563" is ejecting paper rapidly and printing garbage randomly on each of the pages.

SPECIAL NOTE:

The above problems all occurred at about the same time and they seem to all have something in common. The files involved all seem to be on the file system associated with the /usr directory (the lp spooling system heavily uses the /usr directory for its operations)

After checking for other possibilities such as user and/or program errors we finally suspect that the file system associated with the directory /usr might be corrupt.

Step 1: Terminate processes with files open on the suspect file system

Because it is necessary to run `fsck` on a quiescent/unmounted file system, you need to terminate processes with files open on the suspect file system so that you can unmount it. To find out what processes have files open for a given file system, you first need to know which block device file is associated with that file system. You can find this out by entering the command `mount` (with no parameters). This will display a list of currently mounted file systems, which will look similar to this example:

mount

```
/ on /dev/dsk/c0d0s0 read/write on Mon Feb 22 09:09:49 1988
/users2 on /dev/dsk/c3d0s2 read/write on Mon Feb 22 09:09:52 1988
/users on /dev/dsk/c2d0s2 read/write on Mon Feb 22 09:09:54 1988
/usr on /dev/dsk/c1d0s11 read/write on Mon Feb 22 09:09:56 1988
/mnt on /dev/dsk/c0d0s10 read/write on Mon Feb 22 09:09:56 1988
/extra on /dev/dsk/c0d0s9 read/write on Mon Feb 22 09:09:56 1988
/tmp on /dev/dsk/c0d0s3 read/write on Mon Feb 22 09:09:56 1988
```

6

Note

Directories listed in the first field of `mount`'s display represent the top of the tree for their associated file system. To locate which file system you need to check, compare the *full path name* for one of your "problem files" with these directory names. Compare the characters of the two names from left to right until you can no longer match the file's name with one of the entries. The entry that matches the *most* characters represents the file system that contains the "problem file." In the above listing, the directory `/bin` would be on the root file system (the one represented by directory `/`). The file `/mnt/abc.file` is located on the file system represented by the entry `/mnt`.

The block device associated with a file system is listed after the first "on" in each line. In this example, the file system associated with the directory `/usr` has a block device name of `/dev/dsk/c1d0s11` (`/usr` on your system may be in a different location).

Once you have the block device file name for the file system in question, use the command `fuser` to determine what processes (if any) have files open on that file system. In our example, the command and its output would look like this:

```
fuser -u /dev/dsk/c1d0s11
```

```
/dev/dsk/c1d0s11: 76c(root)  35(root)  52(root)  82c(lp)  
                  2533c(lp)  2511c(lp)  2512c(lp)
```

Note In a cluster, this must be done on the cluster server.

The `-u` option tells `fuser` to display in parentheses the name of the user who started the process (in addition to displaying the process ID numbers) for each process with a file open on the specified file system. This enables you to notify those users that you are about to terminate their processes. Once you are sure it is ok to terminate the processes, you can run `fuser` again with the `-k` option to *kill* the processes. For our example, the command would look like this:

```
fuser -k /dev/dsk/c1d0s11
```

`fuser` will display a list of the processes as it kills them.

Note `fuser` uses the SIGKILL signal (equivalent to using the command `kill -9` for each of the processes it kills). This is an immediate and unconditional kill of each process. It doesn't provide the processes a chance to do normal termination processing/cleanup. If it is at all possible to terminate the processes through "normal" procedures (such as using the command `lpshut` to terminate the line printer spooling system's scheduler), you should do so.

Step 2: Unmount the suspect file system

When there are no processes with files open on the suspect file system, unmount it with the `umount` command. For example:

```
umount /usr
```

If `umount` succeeds you should get another prompt. If you get a message indicating “Device Busy”, there are still processes running with files open on the file system.

Note

The root file system is a special case. It *cannot* be unmounted. To safely check the root file system, you should use the `shutdown` command (with no parameters) to switch HP-UX to the single user runstate. This will terminate user processes (and logins) and leave the system console as the only active terminal. You may then proceed with Step 3 (below).

An alternate way to safely check the root file system is to use a *different* file system as the root file system (either by booting your computer from the other file system or by connecting your disk drive to another computer as an auxiliary disk). As long as the file system is not *active* as your root file system (that is, you have *not* booted from it) you can treat it as any other file system. You can then proceed with Step 3 (below).

Step 3: Run fsck using preening mode (option -p or -P)

Once you are *sure* your file system is unmounted (unless it is the root file system), run **fsck** using “preening mode”. **fsck**’s preening mode fixes many file system problems, but never removes data. It is invoked with the **-p** option. Preening mode is *non-interactive*. If **fsck** encounters a problem while running in this mode, it will terminate and request to be run in one of its interactive modes. To run **fsck** against our example file system (using preening mode), you would use the command:

```
fsck -p /dev/dsk/c1d0s11
```

If **fsck** finds no uncorrectable errors, it will print a line of statistics about the file system it checked, indicating its successful completion. The line would look similar to this (with the appropriate statistics for your file system):

```
6737 files, 127283 used, 334693 free (393 frags, 83575 blocks)
```

Use the following table to determine what to do next based on the outcome of your run of **fsck**. For purposes of our example we will assume the worst case (**fsck** encounters uncorrectable errors and requests to be re-run interactively).

fsck outcome	Proceed to Step #
fsck reported no errors	Step 4a
fsck reported errors but corrected them. fsck did <i>not</i> request to be rerun in interactive mode	Step 4b
fsck reported uncorrectable errors and requested to be rerun in interactive mode	Step 5

Step 4a: Check for other causes

If `fsck` completed without encountering any errors, you can be confident that your problem is not a corrupted file system. At this point you need to reexamine other possible causes. Here are a few things that can cause problems with files. There are others; these are the most common.

- A user deleted, overwrote, moved or truncated the file(s)
- A program/application deleted, overwrote, moved or truncated the file(s)
- The *file system* associated with a particular directory at the time a file was created might not be the one that is mounted to that directory at this time (if any are).
- A file (or group of files) was placed in a directory that *now* has a file system mounted to it. The files in the directory prior to the mounting of the current file system *still exist*, but won't be accessible until you unmount the file system that is covering them.
- The protection bits on the file don't permit you to access it
- The ownership of a file does not permit you to access it
- The file has a different name than you thought/entered (remember that HP-UX is case sensitive)
- The file has always been in a different location than where you are looking

Because your file system is *not* corrupt, do not continue with the remaining steps in this procedure (it might be necessary to restore the missing or corrupted files from a backup). This list of possible causes should help you to locate the actual cause of your problem. Record those things you try and what happens so that if it becomes necessary to get help, you can save duplication of your efforts.

Step 4b: Restore any necessary files

Because `fsck` found and corrected errors in the file system, it is possible that this was the cause of the problems you are experiencing. Now that `fsck` has repaired the damage, the file system is once again structurally sound. `fsck` has *not* removed any data (if that would have been necessary, `fsck` would have terminated and requested to be rerun interactively). If any of your files have been lost, it is most likely due to some other cause. For possible causes, refer to the list in Step 4a. If you need these files you will need to restore them from a backup. You are now finished; do not continue with Step 5.

Step 5: Run fsck interactively

`fsck` terminated and requested to be rerun interactively. This indicates that it needs to perform an action that could cause the loss of data or the removal of a file/filename (such as when two files claim ownership of the same data blocks). Because of this, any backups of this file system at this point are likely to fail also. If you have critical files on this file system that have not yet been backed up (and are not yet destroyed), move them to another file system or try saving *only critical files* to tape.

In our example, after reporting and correcting many errors, `fsck` terminated and requested to be rerun interactively.

```
fsck -p /dev/dsk/c1d0s11
/dev/dsk/c1d0s11: BAD DIRECT ADDRESS, SHOULD BE ZERO:
    inode.di_db[1] = 41 (CORRECTED)
/dev/dsk/c1d0s11: BAD DIRECT ADDRESS, SHOULD BE ZERO:
    inode.di_db[2] = 40 (CORRECTED)
/dev/dsk/c1d0s11: 39 DUP I=5
/dev/dsk/c1d0s11: UNEXPECTED INCONSISTENCY;
    RUN fsck MANUALLY.
```

To run `fsck` interactively, leave off the “-p” in the runstring, like this:

```
fsck /dev/dsk/c1d0s11
```

As `fsck` runs into the problem areas, it will request permission to perform certain tasks. If you do not give `fsck` permission to perform the correction, it will bypass the operation, leaving the file system unrepaired. For a list of the errors `fsck` can encounter and what your response to these means, refer to Appendix A.

You can save time by using `fsck`'s `-y` option. When you run `fsck` interactively, if it locates an error, `fsck` will ask you if it should fix the problem. If there are a lot of problems on the disk, a lot of your time could be consumed by answering "y" to every question. `fsck` has an option (the `-y` option), which allows you to tell `fsck` (at the time you run it) that you want it to assume a "y" (yes) answer to ALL questions it will ask.

Caution `fsck`'s `-y` option should *ONLY* be used after `fsck`'s preening mode has requested that you run `fsck` interactively. In interactive mode (or when using the `-y` option), it is possible for `fsck` to destroy data during its repairs. Sometimes this is unavoidable as the only other option would be to leave the file system unrepaired. `fsck`'s preening mode will *only* fix problems that can be fixed *without* losing data. `fsck` also has a `-n` option that will assume an "n" (no) answer to all the questions it would ask interactively. The `-n` is *SAFE* to use because `fsck` will *not* alter the file system. You can use the `-n` option to preview the damage before attempting file system repair.

Step 6: Examine files in the `lost+found` directory

Once you've allowed `fsck` to repair the file system, mount the file system and check its `lost+found` directory for any entries that might be present.

```
mount /usr /dev/dsk/c1d0s11
ls /usr/lost+found
```

If there are any entries present, their names represent their inode numbers. These are files that were “orphaned” (lost association with ANY directory). Examine these files to determine their proper location and name; then return the files to that location. To do this, begin by using the `file` command to determine what *type* of files these are. If they are *ASCII text* files, you can simply list them using `cat` or `more` to see what they contain. If they are some other type, you will have to use a utility such as `xd` or `od` to examine their contents or run the commands `what` or `strings` to help you find the origin of your `lost+found` files.

Important The `lost+found` directory should be empty before you run `fsck` again.

Once you have returned the files in the `lost+found` directory to their proper locations, restore any files you are missing from your most recent backup and you are finished.

Problems with mounting and unmounting file systems

Here are some things to check if you are having problems mounting or unmounting a file system.

Mounting Problems:

- Only *one* file system can be mounted to a directory. If a file system is already mounted to a particular directory and you attempt to mount a second to the same place you, will get an error message indicating “device busy.”
- The device associated with the device file you’re trying to mount is not physically attached or is not in a “ready” state. If you have never mounted this device before, check your device file (`/dev/rdisk/{your device file name}`) to be sure that it has the proper major and minor numbers. If you have a Series 800 Computer, use the command `lssf` to display the location associated with the device file and compare that with the actual hardware address of the disk. For information on device files and major/minor numbers, see *How HP-UX Works: Concepts for the System Administrator*.
- Is the file `/etc/mnttab` present (it should be)? It is normally created (if it doesn’t exist) by the shell script `/etc/rc` when you boot up your computer. If the `/etc/mnttab` does not exist, create it using the following command:

```
/etc/devnm / | grep -v "swap" | /etc/setmnt
```

If `/etc/mnttab` doesn’t exist when you try to mount a file system, you will get an error indicating either that `/etc/mnttab` doesn’t exist or that mount had an “interrupted system call.”

Unmounting Problems

Usually when a file system won't unmount, the problem is that one or more processes have files open on it. Before you can unmount a file system there can be no processes running with files open on it. If processes still have files open on the file system, you will get an error similar to the one in this example:

```
umount /usr                               issue the umount command  
umount: umount(1) of /usr: Device busy     note: cron & the lp spooler  
                                           use /usr
```

To determine which processes, if any, have files open on your file system, use the command `fuser`. For details on how to do this, refer to steps #1 and #2 (only) in the section of this chapter called "Locating and correcting file system corruption".

Where to go for more information

- For more information on creating, mounting and unmounting file systems, see the *System Administration Tasks* manual.
- For detailed information about disk layout, superblocks, inodes, file protection, file sharing, file locking, and file system buffer caching, see *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029.

Disk Space Problems

An overview of disk sectioning

To effectively manage your disk space and handle shortages of disk space, it is necessary to understand how disk space is organized on HP-UX systems. Once you are familiar with this organization, you will know what options are available for resolving disk space shortages, and you will be able to select the best option for your situation.

There are differences in the organization of disk space between the Series 300/400, the Series 700 and the Series 800. These differences are discussed below.

The entire directory tree consists of one or more **file systems**. The primary file system, known as the **root file system** contains the files necessary to get HP-UX up and running. Other file systems can then be mounted to the root file system at strategic points (directories) to provide more disk space along particular directory paths.

Disk organization on Series 300/400/700 computers

On Series 300 computers, there is *only one* file system per physical disk. The file system on each disk will consume all disk space not allocated to the boot and swap areas.

Disk organization on Series 800 computers

On Series 800 computers, disk space can be utilized in two ways:

1. Traditional Disk Sections
2. Logical Volumes

Traditional Disk Sections

Disks on Series 800 computers are divided into chunks of space called **disk sections**. Each disk section can contain one of the following: boot code, swap space or a **file system**. The sections are various sizes and some can be combined (in predefined ways) to create larger disk sections. Figure 7-1 shows, for example, that sections 3,4, and 5 can be combined to form section 8 and that section 2 occupies the entire disk (similar to the Series 300/400/700). You can NOT simultaneously use more than one disk section to reference any given spot on the disk (for example, you can not use sections 3, 4 or 5 if you are using section 8). Not all disk drives have every section shown in Figure 7-1.

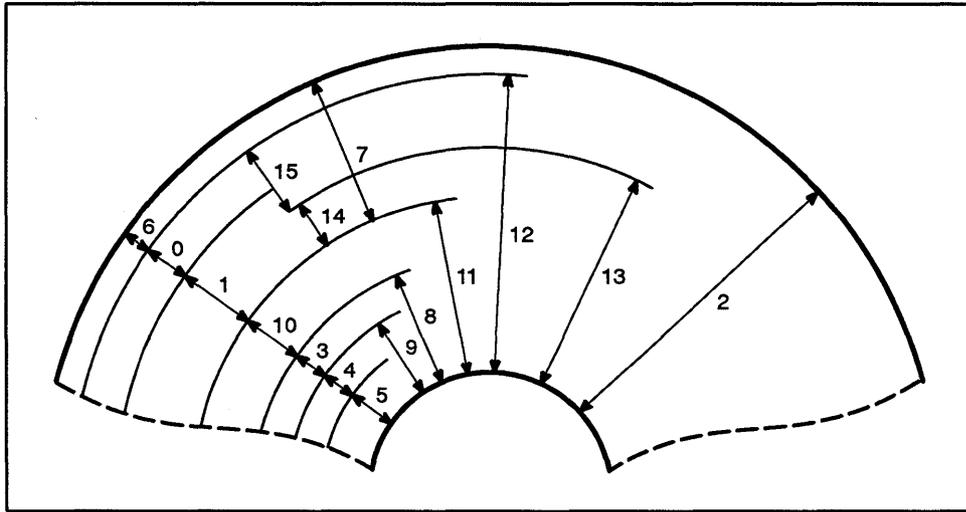


Figure 7-1. Disk Drive Section Map

7-2 Disk Space Problems

Caution

The diagram contained in the `/etc/disktab` file provides a general idea of how sections are arranged on various disks. Some disks vary from this general description. Look closely at the entry in `/etc/disktab` for your specific disk. Some disks, for example, do not have all of the sections shown in the diagram.

For some disk types, sections 0 and 13 overlap, in which case they cannot be used together. Sections 0 and 13 do not overlap—and therefore can be used together—if the sum of their sizes does not exceed the size of section 2.

To find out which sections are available to choose from on your disk drive, consult the entry in your `/etc/disktab` file that corresponds to the model number of your disk drive. There is a manual reference page for `disktab(4)` if you need information on how to read it. At this point, all you need to know is what sections are defined for that drive model. This can be found from viewing the first field of each line in the entry corresponding to your disk drive (for example. `:s0# ...` refers to section 0, `:s6# ...` refers to section 6 etc.).

Caution

If you are using a Series 800 computer, DO NOT EDIT OR CHANGE the `/etc/disktab` file in any way! If you do, this could result in a non-functioning system and you could lose or corrupt the data on your disk!

If you are using a Series 300/400/700 computer, it is possible to add new entries to the file `/etc/disktab`. If (in the unlikely event) you need to do this, follow the directions (located in the file itself) VERY CAREFULLY!

Logical Volumes

With traditional disk sections, you are very limited in how you select and use disk space. Often you are forced to use a disk section that is much larger than you need because it is the only one that is *big enough*. This wastes a lot of disk space that could be used for something else. And, you cannot select a disk section that is larger than a single disk.

The Logical Volume Manager (LVM) subsystem enables a system administrator to manage disk space more flexibly than when using traditional disk sections. Like traditional disk sections, **logical volumes** can hold file systems, swap areas, or raw data. But, unlike traditional disk sections, logical volumes are adjustable in size. Their size can be easily changed to accommodate changing disk space requirements on your system.

Using logical volumes has several other advantages (over traditional disk sections). These are covered in Chapter 8, “Logical Volume Manager (LVM) Problems” in this manual. If you have a Series 800 computer, using LVM is a good way to resolve disk space problems.

Locating the space shortage

A useful command to show you how much free space is available on each of your mounted file systems, is `bdf`.

EXAMPLE:

`bdf`

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/dsk/c0d0s0	23175	20305	552	97%	/
/dev/dsk/c3d0s2	533552	48448	431748	10%	/users2
/dev/dsk/c2d0s2	533552	111100	369096	23%	/users
/dev/dsk/c1d0s11	461976	121088	294690	29%	/usr
/dev/dsk/c0d0s10	123379	11235	99806	10%	/mnt
/dev/dsk/c0d0s9	102513	3613	88648	4%	/extra
/dev/dsk/c0d0s3	27912	80	25040	0%	/tmp

By looking at `bdf`'s output, you can determine which file systems are out or nearly out of space. The disk directories associated with those file systems are listed in the "Mounted on" column of `bdf`'s display.

Note

You might sometimes notice a file system listed as MORE THAN 100% full (in the capacity column). This indicates that the extra disk space normally reserved for superusers only is partially used. See the next section "Accessing the superuser's extra space" for more information on this.

Accessing the superuser's extra space

When a file system is created using the `newfs` or `mkfs` command, a certain amount of disk space (10% unless otherwise specified) is reserved such that, when all other disk space has been used up, only the superuser will be able to write to that file system. For information on how to change the 10% value, see the manual reference pages for `newfs(1m)` or `mkfs(1m)`. 10% is usually adequate; there's rarely a need to use another value.

This space is reserved for performance reasons. Running with less space will significantly degrade performance!

As noted above, if the `bdf` command reports that a file system is at more than 100% capacity, this indicates that this reserved space is partially used. This is similar to the little light on the dashboard of a car telling you that you are about to run out of gas. Take whatever steps you can to free up some disk space.

Resolving disk space problems

There are a number of methods available for handling shortages of disk space. Which method(s) you choose depends on your particular situation.

7 Removing files

Often, when file systems fill up, there are a number of files that are no longer needed. They might include old "test" programs, old data files, backup copies of files made "just in case a change to the real one goofs up something", etc. Locate and remove these files (using the `rm` command) if you are *sure* that they are no longer needed. All users of that file system should do the same.

When programs fail, they sometimes "core dump." As the name implies, a file named `core` is dumped onto the disk (usually in the working directory of the user executing the program). Files named `core` are therefore candidates for removal. Just be sure they don't contain important data and that someone isn't planning to use them for debugging with the `cdb` and `adb` debuggers.

Moving files

If you have file systems that are at or near capacity and others with plenty of space available, you might be able to move files or directories out of the full file system into the lesser used file system. This will even out your disk usage and in some cases might even improve system throughput.

Be careful when moving files to new locations. There are certain files that must reside in a particular place because HP-UX or other software expects them to be there. Most of those files that HP-UX is looking for will reside in one of the directories on the root file system. Your applications might not have requirements for the location of certain files. You should check the documentation for those products for more detail before moving files they may use.

There is a way to *physically* move a file without *logically* moving it. See the section later in this chapter on creating symbolic links.

Archiving files

A combination of the above two methods is to archive files to external media (such as tape or disk pack) that you would like to keep around but will not need to access immediately. This will free up space on your disk for your more immediate needs.

There are several utilities for archiving files to tape and, of course, retrieving them from tape. Among them are `fbackup/frecover`, `cpio` and `tcio`, `dump/restore`, `dd`, and `tar`. Several of these can also be used for moving files/data between disk drives. For information on how to archive files, see the chapter called “Backing Up and Restoring Your Data” in the *System Administration Tasks* manual.

Creating symbolic links

As mentioned above, it is possible to *physically* move a file without *logically* moving it. This is done through a process called “linking.” Links are essentially pointers from the place where the file’s name resides to the place where its data resides (under a different name). There are several types of links, but the type we’re interested in using here is the symbolic link. This

is because symbolic links are the only type that can reference data across file system boundaries.

The file's data can be physically moved to another disk or file system while its name remains where it used to be. The file can then be referenced using the original name by creating a symbolic link to the new location of the data.

There are drawbacks to using symbolic links that you should know about. By creating the link, you are creating a dependency that the file system and file that the link is pointing to will be mounted and available when you need to access the file. Someone might remove, move, or alter the file containing the data without realizing that the link exists. You might end up with a link pointing to nowhere. When using symbolic links, remember that you've created the links and protect the file containing the data accordingly.

For more information on creating symbolic links, consult the manual reference page *cp(1)* (see the *-s* option on the *ln* command).

Shortening files that grow without bound

There are several files on the system that grow without bound. It is necessary to keep an eye on these and regularly clean them up to prevent them from consuming valuable disk space. The following files are of this type:

- */etc/wtmp*
- */etc/btmp*
- */etc/utmp*
- Some processes use log files to record their activities. Many of the files in the directory */usr/adm* are log files and these can potentially grow very large. Other processes can use the directory */tmp* for this purpose.

Note If your computer is part of an HP-UX cluster, the files *wtmp*, *btmp* and *utmp* (listed above) are context-dependent files (CDFs). Each node in the cluster has its own copies of these files.

Using Logical Volumes

If you are using a Series 800 computer, you have a new way to handle disk space problems, the “Logical Volume Manager (LVM).” The Logical Volume Manager allows you a lot of flexibility in how you divide up your disk space. For complete information about the Logical Volume Manager, see the references listed at the end of this chapter, in the section called “Where to go for more information”.

Adding new disks

It might turn out that your needs for immediate access to online information require you to add additional disk drives to your system (or upgrade to larger capacity drives). Of course you can select this option, but you should try the other things mentioned in this section first.

Preparing for and avoiding disk space problems

One of the best ways of avoiding disk space problems is to perform *regular* file system maintenance. Everybody using the system should be in the habit of cleaning up files no longer needed. Doing this on a regular basis will help prevent disk space problems from occurring. Sometimes this type of problem is unavoidable, even with your best efforts. At least you will be able to “see problems coming” and be able to handle them before things reach a crisis stage.

A subsystem called “disk quotas” can also help you control disk usage by the users on your system. For information on using disk quotas, see the references listed at the end of this chapter, in the section called “Where to Go for More Information.”

Where to go for more information

- For information on using disk quotas to monitor and control disk space usage by your users, see Chapter 6, “Managing the File System” in the *System Administration Tasks* manual.
- For information on using the Logical Volume Manager to manage your disk space see:
 - “Managing Logical Volumes” in the *System Administration Tasks* manual
 - Chapter 9, “Logical Volume Manager” in the manual *How HP-UX Works: Concepts for the System Administrator*
- For information on how to handle LVM-related problems, see Chapter 5, “System Boot-Up Problems” in this manual.

Logical Volume Manager (LVM) Problems

With traditional disk sections, you are very limited in how you select and use disk space. Often you are forced to use a disk section that is much larger than you need because it is the only one that is *big enough*. This wastes a lot of disk space that could be used for something else. And, you cannot select a disk section that is larger than a single disk.

The Logical Volume Manager (LVM) subsystem enables a system administrator to manage disk space more flexibly than when using traditional disk sections. Like traditional disk sections, **logical volumes** can hold file systems, swap areas, or raw data. But, unlike traditional disk sections, logical volumes are adjustable in size. Their size can be easily changed to accommodate changing disk space requirements on your system.

Using logical volumes has several other advantages (over traditional disk sections). Here is a list of the advantages that the LVM provides:

- Logical volumes can span disks. A logical volume can be larger than a single disk.
- Logical volumes are adjustable. The size of a logical volume can be adjusted as your system needs change.
- Logical volumes can provide high data availability through the use of **mirroring**. An optional product called MirrorDisk/UX, allows LVM to transparently make additional copies of data in logical volumes. If a mirrored disk fails, other (still working) disks allow you to continue working while the failed disk is repaired. You can also take a copy of a logical volume offline (to back it up, for example) while continuing to update the copy of the logical volume that is still online. When you return the offline copy to service, MirrorDisk/UX will automatically update it.

How LVM is Implemented (The LVM Mechanism)

With the use of the Logical Volume Manager comes an additional layer of complexity. It's important to know at least a little about how LVM is implemented, in order to solve (and prevent) problems that arise from its use.

Traditional Disk Sections vs. Logical Volumes

With traditional disk configurations, a single disk is divided into smaller *fixed-size* and *pre-defined* chunks of space called **disk sections** or **disk partitions**. The size and location of these disk sections cannot be changed.

With the Logical Volume Manager, the boundaries dividing different chunks of disk space are movable. Therefore, a special area is reserved at the beginning of each LVM disk to provide HP-UX with pointers to where these chunks of disk space (called logical volumes) are located. It is this special area that is created when you run the `/etc/pvcreate` command. It is this special area that makes a disk a **physical volume**.

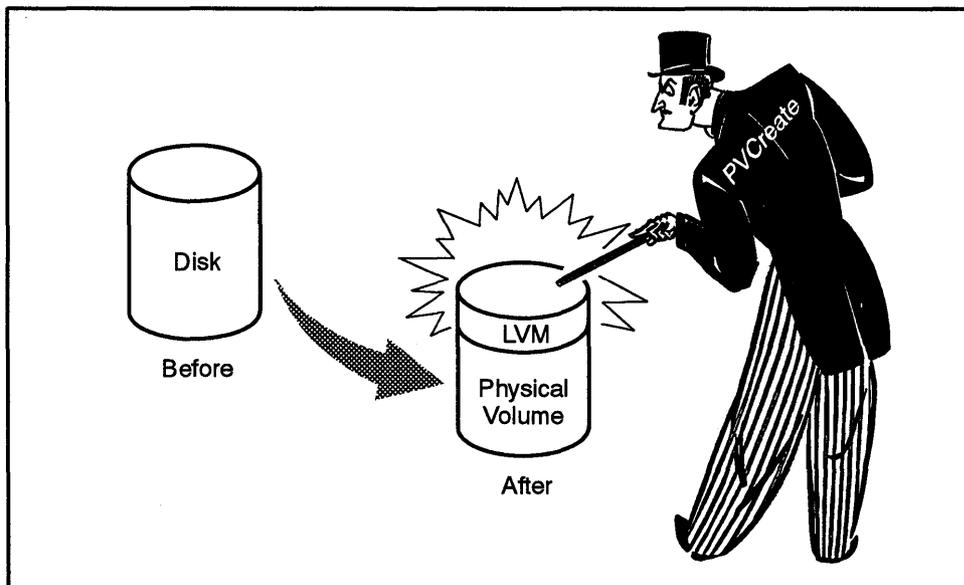


FIG801

Figure 8-1. The pvcreate Command Transforms Disks into Physical Volumes

8-2 Logical Volume Manager (LVM) Problems

Logical Volume Implementation

LVM disks (physical volumes) are sliced up into evenly sized units of disk space called **extents**. Logical volumes are simply groups of extents, allocated from a pool of extents called a **volume group**. The LVM commands (and SAM's logical volume operations) manipulate these extents. Some LVM commands allocate and deallocate extents to and from the various logical volumes. Other LVM commands allow you to add extents to (or remove extents from) a volume group (a whole disk's worth at a time). Still others control the availability of volume groups and the logical volumes within them.

- You can have one or many logical volumes in a volume group, and you can have one or several volume groups on your system.
- You can have some of the disks on your system be traditional disks, and have others be LVM disks.

Quorum

A copy of the LVM configuration information is maintained on each LVM disk (physical volume). The Logical Volume Manager guaranties that it is operating with accurate LVM information (status information, configuration information, etc.) by making sure that a **quorum** of physical volumes contains identical copies of the information.

- When you are booting your system or activating a volume group, a quorum is defined as *more than half* of the physical volumes that are defined for the volume group.
- Once a volume group is active, a quorum is defined to be *at least half* of the physical volumes that are defined for the volume group.

If a quorum of disks is not present when you attempt to activate a volume group, the volume group will not be activated, and its logical volumes will not be accessible. See “Cannot Activate a Volume Group”, later in this chapter, for more information on this condition.

If a quorum of disks is not present when you are booting your system (from a root volume group), your system will not boot. See “Problems with Missing Physical Volumes During Boot”, later in this chapter, for more information on this condition.

If a quorum is lost while a volume group is active (less than half of the physical volumes defined for the volume group continue to be active), your volume group will remain active; however, a message will be printed to the console indicating that the volume group has lost quorum. This console message will include information (the minor number of the volume group) about which volume group lost its quorum. Until the quorum is restored (at least half of the LVM disks in the volume group are once again online), some of the I/O accesses to the logical volumes for that volume group may hang because the underlying disks are not accessible. Also, until the quorum is restored, the **Mirror Write Cache (MWC)** will not be updated because LVM cannot guarantee the consistency (integrity) of the LVM information.

We recommend that you do not make changes to the LVM configuration for active volume groups that do not have a quorum of disks present.

There are ways to override quorum requirements at volume group activation time, or at boot time. These will be discussed later in this chapter. However, the preferred way (the recommended way) to correct this problem is to return the unavailable disks to duty.

LVM Problems, Prevention and Preparation

The best way to handle problems of any nature is to prevent them from occurring in the first place. However, problems cannot always be avoided. Therefore, it is equally important to be prepared for problems in advance, so that you can more quickly solve them when they do occur.

LVM Problem Prevention

Many problems can be avoided if you remember these important things when you are working with LVM. Some of them will be covered in more detail, later in this chapter.

- LVM does nothing special to the user data space. It merely defines the boundaries of that space. All LVM-related information is stored in the specially reserved area at the beginning of LVM-disks and in a few specific system files.
- Logical volumes and their contents are *independent* of each other. For example, if a logical volume contains a file system, changing the size of the logical volume does not change the *defined* size of the file system within it. This is why *reducing* the size of a logical volume that contains a file system must be done with extreme caution.
- A disk that is being used as an LVM disk (a physical volume) cannot *also* be used as a traditionally sectioned disk. However, on a particular system, you can have some disks be LVM disks and others be traditionally sectioned disks.
- The file `/etc/lvmtab` contains information about how physical volumes are grouped on your system (which volume groups contain which disks). Many LVM commands rely on `/etc/lvmtab`, so it is important not to rename it or destroy it.

- In the past you may have used the `/bin/dd` command to recreate a boot area that has been corrupted (by copying the file `/usr/lib/uxbootlf` to the beginning of the raw device file of a disk). **DO NOT DO THIS ANY LONGER!** The LIF area at the beginning of an LVM problem is located in a slightly different area than for a traditional disk. You can destroy all of the LVM information on a disk if you do this. Use the `/etc/mkboot` command to create a boot area on an LVM disk. For details on making an LVM disk a bootable disk, see “Managing Logical Volumes” in the *System Administration Tasks* manual, HP part number B3108-90005.



- Never use `/bin/dd` to copy a section of a non-LVM disk directly to an LVM physical volume without going through the LVM mechanism. That is, if a disk represented by the device `/dev/dsk/c1d0s2` is an LVM disk (a physical volume), **DO NOT** use `/dev/dsk/c1d0s2` as the output file (or input file) in the `dd` command. If you do, you will overwrite the LVM data structures at the beginning of your LVM disk or overwrite a non-LVM disk with LVM disk structures. If you no longer want to use the destination physical volume as an LVM disk, you must properly remove it from the volume group. See the next item in this list.

8

DO NOT:

`/bin/dd if=non-LVM-disk-section of=LVM-physical-volume`

You can, however, use a logical volume as a destination

YOU CAN:

`/bin/dd if=non-LVM-disk-section of=logical-volume-name`

As when working with traditional disk sections, the destination must be as big as the source, and you must take care to not overwrite something critical (such as your root file system or an active swap area).

- Once a disk has been made a physical volume (and is part of a volume group) *never* simply disconnect it from your system and remove it, even if you no longer need the data on the disk. There are LVM commands that you must use to remove the LVM disk from its volume group.

Preparing for LVM Problems

In the same way that you prepare for data loss by doing file system backups, there are several things you can do to prepare to handle LVM problems.

Backing up and Restoring LVM Configuration Information

The most important thing you can do to protect your LVM configuration information is to back it up.

Regular file system backups do not save the LVM configuration information (stored at the beginning of each LVM disk) because this information is not located in a file system. It is in the system's data space on the disk rather than in the user data space. If something happens to the LVM data on your disk (even though the rest of the disk remains intact), HP-UX will not be able to locate the data on the disk. If you can restore the LVM configuration information, you can regain access to your data on that disk; otherwise, you might never again be able to access that data.

Unlike a full file system backup, backing up your LVM configuration data is quick and easy to do. And, it can save you *lots* of time when you are resolving LVM problems.

There are two commands to allow you to save (and restore) your LVM configuration information, `/etc/vgcfgbackup` and `/etc/vgcfgrestore`.

LVM Configuration Data (Backing Up). Perform this procedure once, and repeat it any time you add disks to (or remove disks from) a volume group, add a logical volume, remove a logical volume, change the size of a logical volume, etc.

1. Make sure that all LVM disks (physical volumes) are on line before you make the backup.

2. Create the LVM configuration backup by using the command `/etc/vgcfbackup` for each volume group on your system.

If you are repeating this procedure because you have changed a volume group, you need only do the `vgcfbackup` for the volume groups that have changed since the last time you performed this procedure.

EXAMPLE:

To back up the LVM configuration information for all of the disks in the volume group `/dev/vgroot`:

```
/etc/vgcfbackup /dev/vgroot
```

This saves the information to a file called `/etc/lvmconf/vgroot.conf`. The directory `/etc/lvmconf` will contain all of the `vgcfbackup` data (unless you override this default destination).

3. Once you have backed up all of the configuration data, make a special backup tape (using your favorite backup utility) to back up all of the files in the `/etc/lvmconf` directory and the `/etc/lvmtab` file.
4. Store your tape in a safe location and remember to update it any time you change your LVM configuration.

LVM Configuration Data (Restoring). Here are some common mistakes that will probably require you to restore your LVM configuration information:

- You created a new file system (using the `/etc/newfs` command), but used the device file corresponding to the physical volume that a logical volume resides on, instead of the device file for the logical volume itself.
- You used `/etc/newfs` to create a file system in a disk section on a disk that has been made a physical volume. This is similar to the previous item. Remember that you cannot use a disk as both an LVM disk (physical volume) and a traditionally sectioned disk.
- You used the `/bin/dd` command to copy data to an LVM disk (physical volume) that wiped out the LVM data structures.
- Your `uxgen` input file (S800) sets the dump device such that dumps are dumped on top of LVM information.

Several of these “mistakes” are the result of performing a task that is normally safe to do with traditional disk sections. Operations with LVM sometimes change how you must perform a task.

To restore LVM configuration information that has been destroyed by one of the above actions (or a similar action), do the following:

1. Be sure that the damaged disk is connected to your system and powered on.
2. Use the `vgcfgrestore` command to write the configuration information (previously saved by the `vgcfgbackup` command) to the damaged disk drive.)

EXAMPLE:

```
/etc/vgcfgrestore -n /dev/vgprog /dev/rdisk/c2d0s2
```

If your system is not bootable, see “Can’t Boot From a Logical Volume”, later in this chapter.

Creating a “Picture” of Your LVM System Configuration

Because logical volumes (the LVM equivalent of a disk section or partition) can be part of a disk, part of several disks, an entire disk, several entire disks, singly and doubly mirrored (or not mirrored at all), used for file systems, swap space, or raw data areas, and altered in size, it is more important than ever to have a complete and accurate view of how your system is configured (how the various logical volumes and disk sections are being used).

An accurate picture of how your system is configured will help you to avoid (and more quickly solve) LVM problems. This will be especially helpful if someone who is unfamiliar with your system needs to administer it in your absence.

Your picture should include the following information:

- Which disks on your system are being used as traditional disks (non-LVM disks).
- For each disk drive on your system, record:
 - Disk Model Number
 - Disk Interface Type (SCSI, HP-FL, HP-IB)
 - Disk Device Address (SCSI address, etc.)
 - Disk Capacity
- A list of all the volume groups on your system (and what they're being used for)
- For each volume group on your system, record:
 - The LVM Disks (physical volumes) that are part of that volume group.
 - The logical volumes that are defined for that volume group.
 - How much free (unallocated) space remains in the volume group (and any plans you might have for the use of that space in the future).
- On LVM disks: For each logical volume on your system, record:
 - The logical volume name (for example, /dev/vg01/lvol1)
 - The logical volume size (in megabytes)
 - What the logical volume is being used for (for example, file system /usr, secondary swap, database raw I/O)
 - Whether or not it is mirrored, and if so, where it is mirrored to
- On non-LVM disks: For each disk section in use on your system, record:
 - The section number (for example disk lu 4, section 13)
 - The section size (in megabytes)
 - What the section is being used for (for example, file system /projects)

For a detailed procedure (with examples) on how to create this picture of your system, see the section, "How to Create the Picture of Your LVM System Configuration", later in this chapter.

8-10 Logical Volume Manager (LVM) Problems

The Problem Areas

Can't Boot From a Logical Volume

If you have LVM disks on your system, but your root file system, primary swap area, and dumps area are on traditional disk sections (you don't have a root logical volume), see Chapter 5, "System Boot-Up Problems", in this manual, for how to resolve your bootup problem.

When you are booting from a traditional (non-LVM) disk, the general sequence of events is:

1. Boot ROM initializes hardware
2. Boot ROM locates, loads, and runs the Initial System Loader (ISL) which, in turn loads the HP-UX specific loader (hpux).
3. hpux locates, loads, and runs, HP-UX. hpux is told which disk and disk section contain the kernel to boot, and what that kernel's name is (usually /hp-ux).
4. HP-UX begins running, initializes all of the hardware, locates its disk partitions for its root file system, primary swap area and dump area, and starts up its first processes.

hpux (the ISL utility) and HP-UX (the operating system) know where to locate the disk sections that they need because these sections are a fixed size and at a standard location.

With LVM disks, the general sequence is the same, however the sizes of logical volumes that are being used for the root, swap, and dump areas are configurable for each system. While the root file system must start at the top of the disk, the locations of the primary swap area and dumps area are not preset, and the lengths of all three areas can vary. Therefore, special pointers need to be provided so that these areas can be properly located.

The pointers to the root file system, primary swap area, and dumps area are located within the special LVM data area at the beginning of each bootable LVM disk, along with information about the size of each of these areas.

If you cannot boot from a logical volume, a number of things could be causing the problem. In addition to the problems with non-LVM boots, the following things can be causing an LVM-based system not to boot:

- The LVM pointers at the beginning of the boot disk are absent, corrupted, or simply not current (for example, if you have made a change that affects the size or location of the primary swap area, or the size or location of the dumps area).
- The system thinks it is trying to configure a root, swap, or dumps area on a logical volume, but the disk it is attempting to use is not an LVM disk.
- The system thinks it is trying to boot from a disk partition that has LVM information on it.
- Not enough disks are present in the root volume group to make a quorum. At boot time, you will see a message indicating that not enough physical volumes are available.

As when booting from traditional disk sections, it is helpful to know where in the boot sequence the problem is occurring. If you're not sure and would like help in determining where in the boot sequence the failure is occurring, see Chapter 5, "System Boot-Up Problems" in this manual.

Problems with Missing Physical Volumes During Boot

If there are not enough disks present in the root volume group to constitute a quorum, a message indicating that "not enough physical volumes are present" will be displayed during the boot sequence.

To have a quorum, you need to have present *more than half* of the disks in the volume group. You might see this problem if you originally had one disk in your root volume group, added a second one, and tried to boot the system with one of the two disks not present. Because the volume group has two disks and you must have *more than half* of the disks present, *both disks* in the volume group must be present to boot. Try booting with both disks present (turned on).

If you believe that more than half of the disks in your root volume group are accessible, then you probably added a disk (or several disks) without rerunning the `lvlnboot` command to update the boot data structures. Because at boot time there is no access to the `/etc/lvmtab` file, the hardware paths for *all* of

the physical volumes in the root volume group are kept on each of the boot disks.

If you add new physical volumes to, remove physical volumes from, or otherwise change the configuration of the root volume group, you need to update all of the physical volumes in the group with the new information. Particularly, any time you use the `vgextend`, or `vgreduce` command with the root volume group, you must update the LVM information.

To update the LVM information for the root volume group run the `lvlnboot` command using the “-R” option.

EXAMPLE:

Adding physical volume `/dev/dsk/c1d0s2` to the root volume group `vg00`:

```
vgextend /dev/vg00 /dev/dsk/c1d0s2  
rm -f /dev/root  
lvlnboot -R /dev/vg00
```

Note

If the device file `/dev/root` exists, remove it before running the `lvlnboot` command (with the “-R” option).

If you are sure that more than half of the disks in the root volume group are turned on and properly connected to your system, and you still cannot boot because “not enough disks are present”, then you will need to boot the system overriding the quorum requirement. To do this:

1. Boot your system using the quorum override option (“-lq”) in the “hpux boot” command. For example:

```
hpux -lq (;2)/hp-ux
```

2. Use the `vgchange` command to activate your root volume group and reattach any physical volumes that are listed in the file `/etc/lvmtab`, but are not listed (with their hardware paths) in the LVM boot data areas of the disks in the volume group:

```
vgchange -a y /dev/your_root_volume_group_name
```

3. Update the boot area data structures using the `lvlnboot` command:

```
rm -f /dev/root  
/etc/lvlnboot -v -R /dev/your_root_volume_group_name
```

4. Shut down and reboot your system without using the quorum override

Caution

If some of your disks are not accessible (for example, due to a hardware failure), you need to be concerned that the disks that remain available may all have old LVM configuration information and status. This can *only* happen if *none of the disks that are available now were available the last time that the volume group was in use*.

Check the files in the root file system to be sure that they are up to date. If they look OK, you can use the “init 2” command, or (reboot your system) after running the `lvlnboot` command. This will bring your system up in multiuser mode, so that your users can access it.

If your root file system appears to be out of date, then you must restore it. To do this, follow the procedure in the section of this chapter called “Booting an LVM System in Maintenance Mode”.

If the second disk was simply a mirror copy of the first, and you no longer have access to the second disk:

1. Boot the system overriding the quorum requirement (see example above)
2. Use the `lvreduce` command to reduce the number of mirror copies to 0 (the original copy and no additional copies).
3. Reboot the system without the quorum override option.
4. Rerun the `lvlnboot` command:

```
rm -f /dev/root  
lvlnboot -R /dev/your_root_volume_group_name
```

Problems with Corrupted LVM/Boot Area on Disk

If the LVM data structures have been corrupted on the disk that you are trying to boot (or if they simply aren't there), there are several things that you can try.

If you have mirrored your root logical volume, and primary swap logical volume, and if you made the disk(s) that they were mirrored to bootable (that is to say you used `pvcreate` with the “-B” option and installed the boot programs using the `mkboot` command), you can try booting from the hardware path of one of the mirrors. Once the system is booted from the mirror, you can:

- Restore the LVM data structures (on the “broken” physical volume) from a previously made `vgcfgbackup` file. This is why it is important to make such a backup. For help on how to do this, continue reading this section.
- Use `mkboot` to recreate the boot area (LIF volume) contents on the “broken” disk. For help on how to do this, see “Creating/Recreating a Boot Area on a Disk or Logical Volume” in Chapter 5 of this manual.
- Restore any damage to the root file system itself.

You can then try to boot from the newly repaired disk.

If you have not mirrored the root logical volume, you might be able to boot your system in maintenance mode, and restore the LVM data structures from a previously made `vgcfgbackup` file.

Booting an LVM System in Maintenance Mode

Note Maintenance mode is a special way to boot your system that bypasses the normal LVM structures. It is similar to single-user mode in that many of the processes that normally get started are not, and many of the system checks that are normally performed are not. It is intended to allow you to boot your system long enough for you to repair damage to your system's LVM data structures, which should then allow you to boot your system normally.

Whether or not the maintenance mode boot will succeed depends on what else (if anything) on the disk was also corrupted. In order for your boot to succeed, the LIF header and LIF volume must not have been damaged, and the root file system must be intact. To attempt this:

1. Boot your system using a maintenance mode boot.

EXAMPLE:

```
ISL> hpux -lm (;2)/hp-ux
```

Caution When you have booted your system in maintenance mode:

- Do *not* activate the root volume group
- Do *not* switch to multiuser mode (use the “/etc/init 2” command)

You can corrupt your root file system if you do this!

If you cannot even boot your system using a maintenance mode boot, your root file system is corrupt. You can try booting from a different disk, if you have a mirror copy of the root file system somewhere. Otherwise you will probably need to reinstall.

2. It is often the case that a file called “LABEL” is not present in the LIF area of your boot disk. This file is required by LVM, and if it is not present, your system will not boot in its normal manner.

If you were able to boot into maintenance mode, issue the following command to create a minimal “LABEL” file in the LIF area of your boot disk:

```
/etc/mkboot -i LABEL /dev/rdisk/your_boot_disk_name
```

Substitute the name of the character device file that corresponds to your boot disk, in the above command.

3. Once your minimal “LABEL” file has been created, reboot your system into single-user mode:

EXAMPLE:

```
ISL> hpux -is -lq (;2)/hp-ux
```

If the cause of your boot problem was a missing “LABEL” file, your system should now boot into single-user mode. Proceed to Step 4.

If your system still does not boot, it is likely that your LVM data structures are still corrupt. Reboot your system into maintenance mode and use the `/etc/vgcfgrestore` command to restore recent copies of the data structures. When you do this, be *very sure* that you are restoring up to date information. For details on how to do this see “Backing up and Restoring LVM Configuration Information”, earlier in this chapter.

4. Once your system will boot into single-user mode, there are several things that you should do:
 - a. Run the `lvlnboot` command using the “-R” option to recover the *full contents* of the “LABEL” file.

```
/etc/lvlnboot -v -R /dev/your_root_volume_group_name
```

- b. Check that the files in your root file system are up to date. The concern with overriding the quorum requirement at boot time, and with performing a maintenance mode boot, is that the disks you currently have on line are not the disks that were most recently on line (before your boot problem occurred). It is possible that you are working with an out of date root file system.
- c. If the root file system does not appear up to date, you should restore it from a recent backup.
- d. Once your root file system is up to date, use the `/etc/vgcfgrestore` command to restore recent copies of the LVM data structures. Do this for each of the disks in your root volume group. See “Backing up and Restoring LVM Configuration Information”, earlier in this chapter, for details on how to do this. You should be sure that the configuration information that you are restoring is up to date.

5. Reboot your system to multiuser mode, and verify that all of your file systems (particularly those on logical volumes in the root volume group) are current. Restore any files that are not current from a recent backup.
6. Proceed with your regular operations.

LVM Kernel Trying to Boot from Non-LVM Disk

It might occur that you have generated a new kernel, or are using a kernel other than the one you normally use. If you have defined in the `uxgen` input file (usually `/etc/conf/gen/S800`) that the root device is on a logical volume, but are trying to boot the kernel from a traditionally sectioned disk, your kernel probably will not boot.

This situation is not much different than with a non-LVM system when you have generated the kernel to look for the root device on a particular disk section, when in fact it is located on a different section.

The solution is to boot from a working kernel (or disk), and change the new kernel or your disk usage so that the two match.

Cannot Activate a Volume Group

Volume group activation is done automatically during system startup. Unless you manually deactivate a volume group, or a volume group does not get activated because of a failure to meet quorum, you will probably not need to manually activate a volume group.

Whether a volume group is being activated automatically or manually (using the `vgchange` command), problems can sometimes occur that prevent the volume group activation from succeeding.

The primary thing to look for when resolving this type of problem is missing disks. The usual requirement is that a quorum of disks (more than half of the disks defined for the volume group) must be present. Check to be sure that all disks are powered on, connected to the system, etc. If you are using the “-p” option to `vgchange`, *all* of the physical volumes in the volume group must be present to activate the volume group.

If you attempt to activate a volume group when not enough disks are present to establish a quorum, you will see error messages similar to those in this example:

```
vgchange -a y /dev/vg01
```

```
vgchange: Warning: Couldn't attach to the volume group
                physical volume "/dev/dsk/c1d0s2":
The path of the physical volume refers to a device that does not
exist, or is not configured into the kernel.
```

```
vgchange: Warning: Couldn't attach to the volume group
                physical volume "/dev/dsk/c2d0s2":
The path of the physical volume refers to a device that does not
exist, or is not configured into the kernel.
```

```
vgchange: Couldn't activate volume group "/dev/vg01":
Either no physical volumes are attached or no valid VGDA's were found
on the physical volumes.
```

The highlighting in the above example will not appear on your display. The highlighting and some spacing adjustment are provided to make the example more readable.

To recover from this problem, do *one* of the following things (they are listed in the order of preference):

- Check the power and data connections of all the disks that are part of the volume group that you cannot activate. Return all disks (at least enough to make a quorum) to service. Then, use the `vgchange` command to try to activate the volume group again.
- This problem might occur if you have physically removed a disk from your system (because you no longer intend to use it with that system) but did not remove the physical volume from the volume group. Although you should *never* remove an LVM disk from a system without first removing it from its volume group (using `vgreduce`), you can probably recover from this situation by booting your system with the quorum override option, running `/etc/vgscan` to correct the entry in the `/etc/lvmtab` file, and then rebooting your system with quorum checking enabled.

8

- If there is no way to make a quorum available, the “-q” option to the `vgchange` command will override the quorum check.

EXAMPLE:

```
vgchange -a y -q n /dev/vg01
```

Caution

This will activate the volume group but a quorum will not be present. You might get messages about not being able access logical volumes. This is because some or all of a logical volume might be located on one of the disks that are not present.

Whenever you override a quorum requirement, you run the risk of using data that is not current. Whenever you override a quorum requirement when activating a volume group, be sure to check the data on the logical volumes in that volume group to be sure it is up to date.

You should attempt to return the disabled disks to the volume group as soon as possible. When you return a disk to service that was not online when you originally activated the volume group, use the activation command again to attach the newly accessible disks to the volume group.

EXAMPLE:

```
vgchange -a y /dev/vg01
```

Another possible problem pertaining to activation of a volume group is a missing or corrupted `/etc/lvmtab` file. For information on how to recreate the `lvmtab` file, see “Recovering an `lvmtab` File - Using `vgscan`” later in this chapter.

I Enlarged My Logical Volume: Why Don't I Have More Space?

With traditional disk sections, when you use a section for swap space, the defined swap area is the same size as the section it resides in, and when you make a new file system (for example, using `/etc/newfs`), the file system is the same size as the disk section it resides in. The “container” (the disk section) never changes in size, so you don't have to worry about the size of its contents.

With logical volumes, however, it is important to understand that the “container” (the logical volume) and its contents (a file system, swap area, or raw data storage area) are two distinct things. Increasing the size of the container does *not* automatically increase the size of its contents. Reducing the size of the container can *destroy* part of its contents, which can lead to further problems (such as system crashes).

File systems

When you originally create a file system, whether it is in a traditional disk section or a logical volume, the `newfs` command will create the file system the same size as the *then current* size of the disk section or logical volume. The file system has its own view of how big it is. If you enlarge a logical volume that contains a file system (using the `/etc/lvextend` command), the file system within it does not automatically know that its container has been enlarged. You must tell it that this is so by using the `/etc/extendfs` command.

EXAMPLE:

The current size of a logical volume called `/dev/vg01/lvprog` is 1024 Mb (1 Gigabyte). Because the programmers using the file system in this logical volume have consumed 95% of its current space and a new project is being added to their work load, it needs to be enlarged.

Here is what the file system looks like originally.

Filesystem	kbytes	used	avail	capacity	Mounted on
<code>/dev/vg01/lvprog</code>	1008566	852016	55693	94%	<code>/programmers</code>

Now we use the `lvextend` command to enlarge the *logical volume*.

```
/etc/lvextend -L 1200 /dev/vg01/lvprog
```

Logical volume `"/dev/vg01/lvprog"` has been successfully extended.

Notice that even though we have enlarged the logical volume, the *file system* has not increased in size.

Filesystem	kbytes	used	avail	capacity	Mounted on
<code>/dev/vg01/lvprog</code>	1008566	852016	55693	94%	<code>/programmers</code>

Before the programmers can utilize the extra space in their logical volume, the file system `/programmers` must be extended.

Notice that you must first unmount the file system before you can extend it.

8-22 Logical Volume Manager (LVM) Problems

EXAMPLE:

```
/etc/extendfs /dev/vgtdcb/lvprog  
cannot extend a mounted filesystem /dev/vgtdcb/rlvprog.  
/etc/umount /programmers  
/etc/extendfs /dev/vgtdcb/lvprog  
max number of sectors extendible is 180224.  
extend file system /dev/vgtdcb/lvprog to have 180224 sectors more.  
Warning: 64 sector(s) in last cylinder unallocated  
extended super-block backups (for fsck -b#) at:  
1053808, 1061008, 1068208, 1075408, 1082608, 1089808, 1097008, 1104208, 1111408, 1118608,  
1125808, 1133008, 1140208, 1146896, 1154096, 1161296, 1168496, 1175696, 1182896, 1190096,  
1197296, 1204496, 1211696, 1218896, 1226096,
```

Now that the file system has been increased in size, this is reflected in the output from the BDF command.

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/vg01/lvprog	1181990	852016	211775	80%	/programmers

Swap Space

When you originally enable swapping in a given swap area, HP-UX determines how large the area is and will use no more space than that. If the swap space is in a logical volume and you use the `/etc/lvextend` command to enlarge the logical volume, you must reboot your system before HP-UX will know that it can use the extra space that you have provided.

I Reduced the Size of a Logical Volume and Now My System Crashes!

The distinction between the size of a logical volume and the size of its contents is important to know when you are extending a logical volume; it is *critical* to know when you are *reducing* the size of a logical volume.

File Systems

Caution

If you reduce the size of a logical volume that contains a file system, you could corrupt the file system and potentially crash your system.

File system corruption will occur any time that you reduce the size of a logical volume to a size smaller than that of its file system. When a file system is originally created in a logical volume, the `/etc/newfs` command will make the file system as large as the the logical volume will permit.

Swap Space

The `lvreduce` command will not allow you to decrease the size of an active swap area.

What to do if Your System Crashes

If you have reduced the size of a logical volume that contains a file system such that it is smaller than the size of the file system within it, you have corrupted part of the file system and the data within it. If you then attempt to access the part of the file system that has been corrupted, you will probably panic your system. If this occurs:

1. Reboot your system
2. (Optional step: Read caution below!) If the data in the now corrupt file system is critical and you do not have a current backup of that data, you can *try* to recover the part of the data that remains intact by backing up the files on that file system in your usual way.

Caution

Before you attempt this backup, you must be aware of two things:

- a. When your backup program accesses the corrupt part of the file system, your system will panic again! You will need to reboot your system again to continue with the next step.
- b. Your backup will *NOT* be complete. There is *no guarantee* that all (or any) of your data on that file system will be intact or recoverable. This is merely an attempt to save as much as possible.

-
3. If it is mounted, immediately unmount the corrupted file system.
 4. You can now use the logical volume for swap space or raw data storage, or use the `/etc/newfs` command to create a *new* file system in the logical volume. This new file system will match the current (smaller) size of the logical volume.
 5. If you have made a new file system on the logical volume, you can now:
 - Restore the contents of a previous backup (not the backup from step 2)
 - Use the new file system for a new purpose (no file restoration)
 - *Attempt* to restore as many files as possible from the backup you made in step 2. Again, there are no guarantees that any of the data will be recoverable from this backup.

When it is Safe to Reduce the Size of a Logical Volume

If you extend the logical volume, *but not its file system*, you can safely reduce the logical volume's size as long as it remains big enough to hold its file system. Once you use the `/etc/extendfs` command to expand the file system, you can no longer safely reduce the size of the associated logical volume.

Note

- There is no “reducefs” command. You cannot reduce the size of a file system.
 - If you use SAM to extend a logical volume that contains a file system, SAM will automatically run `extendfs` for you. So, you can no longer safely reduce the size of a logical volume containing a file system, if you extended it using SAM.
-

Program Hangs: No Response or Program Output

You might occasionally see long periods of apparent inactivity by programs that are accessing disks. It will appear as if the programs are hung. In fact they may be hung, waiting for access to a currently inaccessible disk. Messages about the disk being offline will also appear on your system console.

If the logical volume is mirrored onto another disk, the hang will only last a minute or so (while LVM waits for the disk driver to return). When the disk driver returns with “disk offline” error condition, LVM marks that disk as offline, and continues the mirror operation on the other mirror disks. LVM will not continue to attempt to write to the offline disk; rather, it will write to the mirror copy until the offline disk is returned to the volume group.

If the logical volume is not mirrored, or if the mirror copies of the logical volume are also not present, the program will hang until the disk that it is trying to access is accessible.

The solution in these cases is generally to check what has happened to the disk drives on your system and get them back on line as soon as possible.

How to Create the Picture of Your LVM System Configuration

Here is a procedure that you can use to construct a picture of the configuration of a system that you are unfamiliar with. Although the entire procedure is rather long, it is divided into parts that can be done at different times. In other words, you do not have to complete the entire procedure in one session. However, do follow the procedure in sequence. The time you invest in assembling this information can save you a lot of time when you are troubleshooting LVM-related problems.

The examples throughout this procedure are based on a hypothetical system owned by a modern railroad. The system is used for different purposes during different hours of the day. The volume groups on the system are based on the system's different types of users. This system has many disks on it; some are SCSI disks, some are HP-FL disks, and some are HP-IB disks. Your system might not be as complex.

Your goal is to create a "picture" of your system that you can use as a tool in solving LVM-related problems. You do not have to use the same format as shown in this procedure (as long as you include the same information). The important thing to do is to create your "picture" in a form that can be faxed.

If you ever need to call someone (such as HP's Response Center) for help with an LVM problem, you can fax or show the picture (and other information) to whomever is helping you. This will help that person more quickly resolve your problem.

Note Some LVM-related problems, prevent you from booting your system. Therefore, be sure to have a printed copy of this information. If you keep it electronically (on your system) and you can't boot the system, you won't be able to retrieve the information you have gathered.

Part 1: Your System's Hardware

This part of the procedure will help you gather information about the disks and other hardware on your system pertaining to LVM.

1. Identify what type of computer you have using the `uname -a` command.

EXAMPLE:

```
uname -a
```

```
HP-UX Erail1 A.09.00 U 9000/857 1234567890 unlimited-user license
```

2. Look at the interface cards in your computer and (perhaps by tracing cables) identify which cards have disk drives hooked to them. For each card (with disk drives attached to it), identify:
 - a. What slot number it is in
 - b. What type of card it is (for example, HP-FL, HP-IB, SCSI)
 - c. What disk drives are hooked to it (for example, C2470S, hp7963B, etc.)
 - d. The disk drives' addresses (SCSI device address, HP-IB address, etc.)

You might want to include the disk drive storage capacities in your diagram. This information can be found in the manuals that came with your disk drives or by using the `/etc/diskinfo` command.

EXAMPLE:

Table 8-1. Sample LVM System Configuration (Part 1)

System Type = 857				
Slot #	I/F Card Type	Disk	Device Address	Capacity
1	HP-FL	C2201A	0	670 Mb
		C2201A	1	670 Mb
3	HP-IB	hp7963B	0	304 Mb
		C2282A	1	670 Mb
9	SCSI	C2474S	6	1,350 Mb
		C2773S	5	677 Mb
11	SCSI	C2474S	6	1,350 Mb
		C2473S	5	677 Mb
13	SCSI	C2472S	6	422 Mb
		C2472S	5	422 Mb

- Identify the hardware path address for each of the disks on your system. You can do this with the `/etc/ioscan` command shown in the example below:

EXAMPLE:

```
ioscan -fC disk
```

```
# ioscan -fC disk
```

Class	LU	H/W Path	Driver	H/W Status	S/W Status
disk	1	4.0.0	hpfl1.target.disc4	ok(0x0)	ok
disk	2	4.1.0	hpfl1.target.disc4	ok(0x0)	ok
disk	3	12.1.0	hpib1.disc1	ok(0x0)	ok
disk	4	12.2.0	hpib1.disc1	ok(0x0)	ok
disk	5	36.6.0	scsi1.target.disc3	ok(0x0)	ok
disk	6	36.5.0	scsi1.target.disc3	ok(0x0)	ok
disk	7	44.6.0	scsi1.target.disc3	ok(0x0)	ok
disk	8	44.5.0	scsi1.target.disc3	ok(0x0)	ok
disk	0	52.6.0	scsi1.target.disc3	ok(0x0)	ok

- Match the entries in the `ioscan` output with the drives on your list from step 2 (Table 8-1).

To get the slot number of the interface card for each device in the `ioscan` output divide the module number (the number before the first period in the “H/W Path”) by 4.

Note On some of the larger Series 800 computers, the module number might be preceded by a bus converter number (a number followed by a “/”). Do not use the bus converter number by mistake.

8

The other numbers in the hardware path include device address information that can help you identify which disk drive (of all the disk drives on a specific interface card) corresponds to each entry in `ioscan`’s output. These numbers differ depending on whether you have a CIO or an HP-PB based computer, and depending on what type of interface you have (HP-IB, SCSI, HP-FL, etc.). If you are not familiar with how to decode the rest of the numbers in the hardware path, see Chapter 10, “System Architectures” in the manual *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029 for assistance.

EXAMPLE:

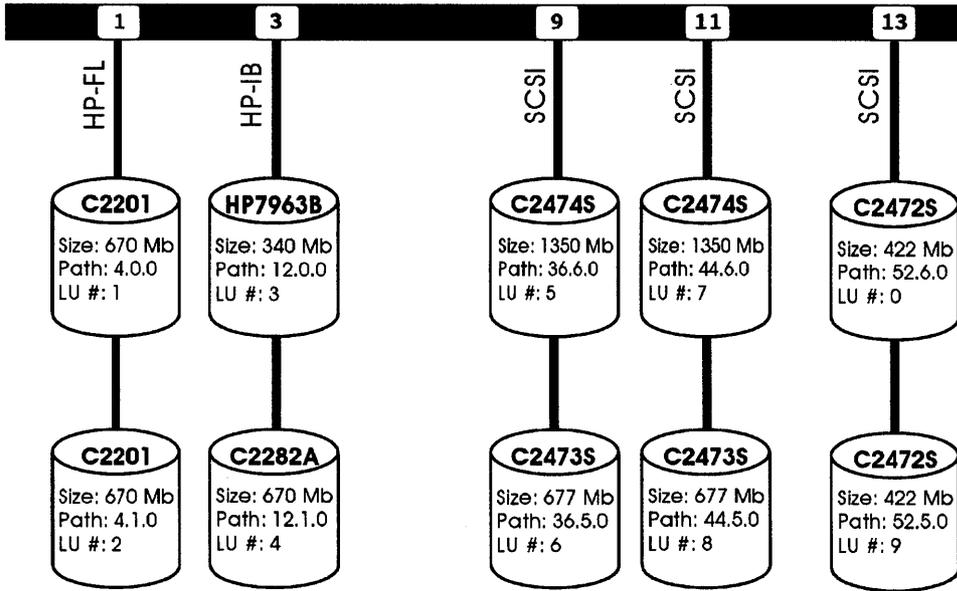
In the above example, the entry:

Class	LU	H/W Path	Driver	H/W Status	S/W Status
disk	8	44.5.0	scsi1.target.disc3	ok(0x0)	ok

indicates that:

- The disk is attached to the interface card in Slot 11 (44 divided by 4)
 - It is a SCSI disk (from Table 8-1, we know that the card in slot 11 is a SCSI card)
 - It's SCSI device address is 5 (from Table 8-1, we know that this disk drive is a 677 Mb, C2473S disk drive)
 - The logical unit (LU) number for this disk drive is 8 (you'll see this number later)
5. From the information about your system that you have gathered so far, create a picture that is similar to Figure 8-2.

SPU Backplane (Slot Numbers)



Computer Type: HP9000 Model 857
 System Name: Erail1
 HP-UX Revision: A.09.00

Figure 8-2. The System's Disk Drives

Part 2: The Volume Groups on Your System

This part of the procedure will help you identify:

- The volume groups on your system
- Which of the disks on your system are associated with each volume group.

1. Begin by executing the following command:

```
/etc/vgdisplay -v | grep -i name | grep -iv lv > /tmp/volumegroups
```

The heart of the above compound command is the `vgdisplay` command. From the `vgdisplay` output, the `grep` commands extract the specific information that you are interested in. The above command will save the information in the file `/tmp/volumegroups`.

EXAMPLE:

For our sample system, the output would look like this:

```
VG Name          /dev/vgroot
  PV Name        /dev/dsk/c0 d0s2
  PV Name        /dev/dsk/c9 d0s2
VG Name          /dev/vgprog
  PV Name        /dev/dsk/c1 d0s2
  PV Name        /dev/dsk/c2 d0s2
VG Name          /dev/vgtcdb
  PV Name        /dev/dsk/c5 d0s2
  PV Name        /dev/dsk/c6 d0s2
  PV Name        /dev/dsk/c7 d0s2
  PV Name        /dev/dsk/c8 d0s2
```

The shaded digits in the above example indicate the logical unit numbers associated with the disk in each “PV Name” entry. “PV” in this output stands for “Physical Volume” (in other words, an LVM disk).

Note In our example, logical unit numbers 3 and 4 are not listed because the disks that are associated with them are not currently part of any volume group.

- Use the logical unit numbers from the file you created in the previous step (`/tmp/volumegroups` in our example) and the output from the `ioscan` command that you executed in part 1 of this procedure (`/etc/ioscan -fC disk`) to identify which disks on your system belong to each of the volume groups. Run the `ioscan` command again if necessary. The column of `ioscan`'s output labeled "LU" represents the logical unit number for the entries it lists.

EXAMPLE:

Entry from our `ioscan` output in part 1:

Class	LU	H/W Path	Driver	H/W Status	S/W Status
disk	5	36.6.0	scsi1.target.disc3	ok(0x0)	ok

From Table 8-1 we know that this is the C2747S disk drive in slot 9 (36 divided by 4) with SCSI device address 6. The LU number for this device is 5.

Now that we know the logical unit number, we can match it with the entry from our `/tmp/volumegroups` file. The entry that matches is:

PV Name	/dev/dsk/c5d0s2
---------	-----------------

And, by looking at the `/tmp/volumegroups` file, we know that the 1,350 Mb disk drive attached to the interface card in slot 9, having a SCSI device address of 6 and a logical unit number of 5, is part of the volume group `/dev/vgtcdb`.

3. Edit the `/tmp/volumegroups` file to record this information. Also include a short description of what the volume groups are being used for. Use whatever format best meets your needs. Here is a suggested format:

```

VG Name          /dev/vgroot          || Use: Root Volume Group
PV Name          /dev/dsk/c0d0s2     || 422 Mb - Slot 13 Address 6
PV Name          /dev/dsk/c9d0s2     || 422 Mb - Slot 13 Address 5

VG Name          /dev/vgprog          || Use: Programmers Volume Group
PV Name          /dev/dsk/c1d0s2     || 670 Mb - Slot 1 Address 0
PV Name          /dev/dsk/c2d0s2     || 670 Mb - Slot 1 Address 1

VG Name          /dev/vgtcdb          || Use: Train-Control Database
PV Name          /dev/dsk/c5d0s2     || 1,350 Mb - Slot 9 Address 6
PV Name          /dev/dsk/c6d0s2     || 677 Mb - Slot 9 Address 5
PV Name          /dev/dsk/c7d0s2     || 1,350 Mb - Slot 11 Address 6
PV Name          /dev/dsk/c8d0s2     || 677 Mb - Slot 11 Address 5

```

The shaded entry represents the one from the example in step 2, above.

4. Print the file that you have created and edited.

This final step in this part of the procedure is very important. You might need this information at a time when your system won't boot or when your system is not available.

Part 3: The Logical Volumes on Your System

In part 2 of this procedure you identified the volume groups that are defined on your system. These are pools of disk space from which logical volumes (the LVM equivalent of disk sections) are created. This part of the procedure will help you identify:

- The logical volumes on your system
 - Which volume groups they are part of
 - How big they are
 - What they're being used for
1. Begin this part of the procedure using a slight variation of the compound command that began part 2. Be sure to use a different destination file name so that you don't overwrite your previous work.

```
/etc/vgdisplay -v|grep -i name|grep -iv pv >/tmp/logicalvolumes
```

The difference between this command and the similar command in part 2 (besides the destination file name) is the second grep command. This time we are filtering out lines with the string "pv" instead of the string "lv".

EXAMPLE:

For our sample system, our output would look like this:

```
VG Name          /dev/vgroot
LV Name          /dev/vgroot/lvol1
LV Name          /dev/vgroot/lvol2
LV Name          /dev/vgroot/lvol3
VG Name          /dev/vgprog
LV Name          /dev/vgprog/lvprog
VG Name          /dev/vgtcdb
LV Name          /dev/vgtcdb/trains
LV Name          /dev/vgtcdb/stations
```

2. For each logical volume in your /tmp/logicalvolumes, use the following command to find its size:

```
/etc/lvdisplay /dev/VGname/LVname |grep -i size
```

EXAMPLE:

To find the sizes of the two logical volumes that are part of the volume group “vgtcdb”:

```
/etc/lvdisplay /dev/vgtcdb/trains | grep -i size  
LV Size (Mbytes)          1705
```

```
/etc/lvdisplay /dev/vgtcdb/stations | grep -i size  
LV Size (Mbytes)          127
```

3. The next step is to determine what each of the logical volumes is being used for. You will need to do several things to determine this.
 - a. Determine which logical volumes you are using for swap space.

Use the `/etc/swapinfo` command to determine which (if any) of your logical volumes are being used for swap space.

EXAMPLE:

```
/etc/swapinfo -m
```

	Mb	Mb	Mb	PCT	START/	Mb		
TYPE	AVAIL	USED	FREE	USED	LIMIT	RESERVE	PRI	NAME
dev	120	53	67	44%	-	-	0	/dev/vgroot/lvol2
hold	0	29	-29					

Look in the last column of the swapinfo output (“NAME”) for device file names that match “LV Name” entries in the `/tmp/logicalvolumes` file that you made in the previous step. These are the logical volumes that you are using for swap space.

- b. Determine which logical volumes you are using for file systems.

Use the `/usr/bin/bdf` command to list the mounted file systems.

EXAMPLE:

```
/usr/bin/bdf
  Filesystem      kbytes   used   avail capacity Mounted on
/dev/vgroot/lvol1 130520  45503  71965    39%      /
/dev/dsk/c3d0s2   310272 117641 161603    42%     /deptA
/dev/dsk/c4d0s2   685056 513074 103476    83%     /deptB
/dev/vgroot/lvol3   92070   53777  38293    58%     /tmp
/dev/vgprog/lvprog 1047552 899789  43007    95%     /programmers
```

Notice in the above example that the file systems `/deptA` and `/deptB` are mounted on *traditional disk sections*. These are the two HP-IB disks (with logical unit numbers 3 and 4).

- c. Determine which logical volumes are being used for raw I/O and those that have currently unmounted file systems.

Unfortunately, there is no way easy to do this. For the logical volumes in your `/tmp/logicalvolumes` file that are not being used for swap or a currently mounted file, you will have to be familiar with the operations on the system in order to determine what these are being used for (if anything at all).

EXAMPLE:

In our modern railroad's system, the logical volumes in the volume group `/dev/vgtcdb` are being used for raw I/O by the railroad's train-control database. From working at the railroad, we just knew that that is what they were using the logical volumes `/dev/vgtcdb/trains` and `/dev/vgtcdb/stations` for.

4. Edit the `/tmp/logicalvolumes` file (the file you created in step 1 of this part of the procedure) to include the information that you have gathered about how the logical volumes are being used on your system and how big they are. You can use any format that best meets your needs. We suggest the following format:

```

=====
      Logical Volume Name          Size of LV      Used for:
=====
VG Name  /dev/vgroot
  LV Name /dev/vgroot/lvol1        127 Mb      Root file system (/)
  LV Name /dev/vgroot/lvol2        205 Mb      Swap Space
  LV Name /dev/vgroot/lvol3         90 Mb      /tmp

VG Name  /dev/vgprog
  LV Name /dev/vgprog/lvprog       1340 Mb     /programmers

VG Name  /dev/vgtcdb
  LV Name /dev/vgtcdb/trains       1704 Mb     Raw I/O Train Control
  LV Name /dev/vgtcdb/stations     124 Mb     Raw I/O Station Control

```

5. Print the file that you have created and edited.

As with the file you created in part 2, it is very important that you print the file you have just created. You might need this information at a time when your system won't boot or when for some other reason it is not available.

Part 4: The Mirrors on Your System

Note You can skip this part of the procedure if you are sure that your system does not have the MirrorDisk/UX product on it.

If you're not sure whether or not MirrorDisk/UX is on your system, you can determine this by checking for the presence of the file `/etc/vgsync`. This file will only be present if you have MirrorDisk/UX on your system.

In this part of the procedure, you need to determine:

- Which logical volumes on your system are mirrored
- How many mirror copies there are of each mirrored logical volume
- Where those mirror copies are located

8-40 Logical Volume Manager (LVM) Problems

- The **mirror allocation policy** being used by each mirrored logical volume
- The consistency recovery mechanism being used by each mirrored logical volume

For details on what each of these things are, see Chapter 9, “Logical Volume Manager” in the *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029.

LVM Mirrors provide extra protection against data loss due to hardware failure. There are different ways to set up mirroring (using different recovery mechanisms) to adjust the level of risk (and the level of system performance). If you are troubleshooting an LVM problem, or recovering from a system crash on a system that has LVM Mirrors, it is important to know which data was mirrored, to where, and what consistency recovery mechanism was being used. These things affect the time it takes to recover from a crash and how confident you can be in your data after the recovery.

Gathering this information is fairly simple.

1. For each logical volume in your `/tmp/logicalvolumes` file (the file you built in part 3 of this procedure, use the `/etc/lvdisplay` command with its “-v” option. Save the output from each of your `lvdisplay` commands to a different file so that you can print these files and keep them with the rest of your information. Choose whatever names you want for the files you create. For each `lvdisplay` command, you will get output similar to this example from our sample system:

EXAMPLE:

```

/etc/lvdisplay -v /dev/vgtdcb/trains
--- Logical volumes ---
LV Name           /dev/vgtdcb/trains
VG Name           /dev/vgtdcb
LV Permission     read/write
LV Status         available/syncd
Mirror copies     1
Consistency Recovery MWC
Schedule          parallel
LV Size (Mbytes)  1704
Current LE        426
Allocated PE      852
Bad block         off
Allocation        strict

--- Distribution of logical volume ---
PV Name           LE on PV  PE on PV
/dev/dsk/c0d0s2   426       426
/dev/dsk/c9d0s2   426       426

--- Logical extents ---
LE  PV1           PE1  Status 1  PV2           PE2  Status 2
0000 /dev/dsk/c0d0s2 0000 current /dev/dsk/c9d0s2 0000 current
0001 /dev/dsk/c0d0s2 0001 current /dev/dsk/c9d0s2 0001 current
0002 /dev/dsk/c0d0s2 0002 current /dev/dsk/c9d0s2 0002 current
      :           :
0424 /dev/dsk/c0d0s2 0424 current /dev/dsk/c9d0s2 0424 current
0425 /dev/dsk/c0d0s2 0425 current /dev/dsk/c9d0s2 0425 current

```

2. To determine which logical volumes are mirrored, and how many mirror copies there are of each mirrored logical volume, look at your output from step 1 of this part of the procedure. The `lvdisplay` output item called “Mirror copies” will have a value of:
 - 0 if the logical volume is not mirrored
 - 1 if the logical volume has one copy of the data *in addition to* the original
 - 2 if the logical volume has two copies of the data *in addition to* the original
3. To determine where the copies of the data are, you need to look at the “Logical extents” entries that make up the last part of the display. In this part of the display, you will see column headings such as these:

```

LE   PV1                PE1  Status 1 PV2                PE2  Status 2

```

There is one line of output for each **logical extent** on the currently displayed logical volume. The number of the logical extent is located in the first column (“LE”).

In the case of a mirrored logical volume, there are two or three *physical copies* of each logical extent. Each of these copies is known as a **physical extent**.

The location, physical extent number, and status of the original (primary) copy of each logical extent are shown in the columns “PV1”, “PE1”, and “Status 1”, respectively. If there are no mirror copies of a logical volume, there would be no columns other than these on the display.

If there is a second physical copy of a logical extent, its location, physical extent number, and status would be located in the columns “PV2”, “PE2”, and “Status 2”, respectively.

Note	If there is nothing listed in the “PV2” column, this indicates that this physical extent is located on the same physical volume as the original copy of the data (the physical volume listed in the column “PV1”).
-------------	--

If there is a third physical copy of a logical extent, its location, physical extent number, and status would be located in the columns “PV3”, “PE3”, and “Status 3”, respectively.

Note

When you have three physical copies (original plus two mirror copies) of a logical extent, the output becomes a bit difficult on an 80 column display. If you have a printer that can print wide lines (around 115 characters per line), printing the `lvdisplay` output can make it easier to read. If you are viewing the output in an X-window, try widening the window to prevent the line-wrap of the extra characters.

EXAMPLE:

In the output for the logical volume `/dev/vgtdcb/trains`, shown in step 1, the primary copy of each logical extent is located on the physical volume `/dev/dsk/c0d0s2`. There is an additional copy of each logical extent (a second physical extent) located on the physical volume `/dev/dsk/c9d0s2`.

4. The mirror allocation policy for a logical volume is listed in `lvdisplay`'s output in the field called “Allocation”. Its value will be “strict”, “PVG-strict” or “non-strict”. This has to do with where LVM will put additional copies of a mirrored logical extent. For specifics about what each of these mean, see Chapter 9, “Logical Volume Manager” in the manual *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029.
5. The consistency recovery mechanism for a logical volume is listed in `lvdisplay`'s output in the field called “Consistency recovery”. Its value will be “MWC”, “NOMWC”, or “NONE”. This has to do with how LVM performs mirror consistency recovery during the activation of the volume group containing this logical volume.
6. As with the previous parts of this procedure, we strongly recommend that you print out the information you have gathered during part 4, that is, the `lvdisplay` output for each of the logical volumes on your system.

Part 5: Putting it all Together (Presenting the Picture)

If you've worked through this procedure, you have gathered a *lot* of information. The purpose of gathering all of this information is to create a picture of how your LVM-based system is configured. This picture will serve as a tool to help you solve LVM-related problems. And, you can copy, fax, or show the picture to anyone trying to assist you with an LVM-related problem (such as Hewlett-Packard's Response Center engineers).

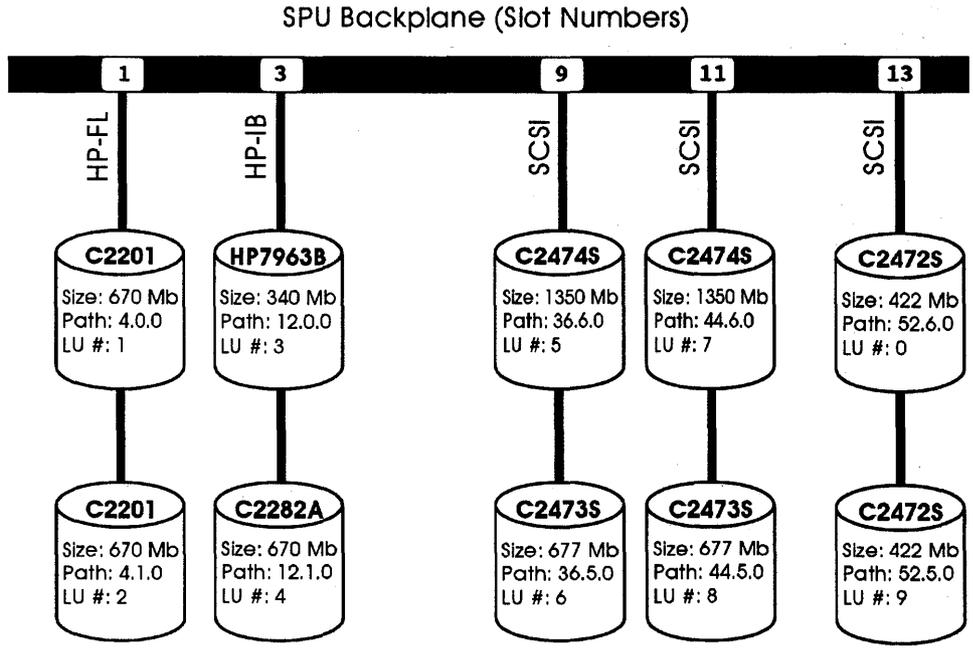
The final part of this procedure is to put the information that you have gathered in an easily readable form, so that you can refer to it when problems arise.

There is no required format for this, but here is a suggestion based on the sample system that we have been using throughout this procedure.

Whatever format you choose, remember to:

- Keep a hardcopy of your system's configuration
- Keep the information up to date by adjusting it as necessary any time that you:
 - Add a new disk to your system (especially if it will be an LVM disk)
 - Remove a disk from your system
 - Create new logical volumes
 - Adjust the size of logical volumes
 - Remove logical volumes
 - Change how you are using a volume group or logical volume
- Tell somebody else where this information is, in case someone else needs to assume your responsibilities unexpectedly.

Here is the picture that we created during part 1 of this procedure.



Computer Type: HP9000 Model 857
 System Name: Erai1
 HP-UX Revision: A.09.00

Figure 8-3. The System's Disk Drives

8

Here are printouts of the two files that we created in parts 2 and 3 of this procedure: the information about the volume groups and logical volumes on our system.

CONFIGURATION OF PHYSICAL VOLUMES:

```

VG Name          /dev/vgroot          || Use: Root Volume Group
PV Name          /dev/dsk/c0d0s2      || 422 Mb - Slot 13 Address 6
PV Name          /dev/dsk/c9d0s2      || 422 Mb - Slot 13 Address 5

VG Name          /dev/vgprog          || Use: Programmers Volume Group
PV Name          /dev/dsk/c1d0s2      || 670 Mb - Slot 1 Address 0
PV Name          /dev/dsk/c2d0s2      || 670 Mb - Slot 1 Address 1

VG Name          /dev/vgtcdb          || Use: Train-Control Database
PV Name          /dev/dsk/c5d0s2      || 1,350 Mb - Slot 9 Address 6
PV Name          /dev/dsk/c6d0s2      || 677 Mb - Slot 9 Address 5
PV Name          /dev/dsk/c7d0s2      || 1,350 Mb - Slot 11 Address 6
PV Name          /dev/dsk/c8d0s2      || 677 Mb - Slot 11 Address 5

```

CONFIGURATION OF LOGICAL VOLUMES:

```

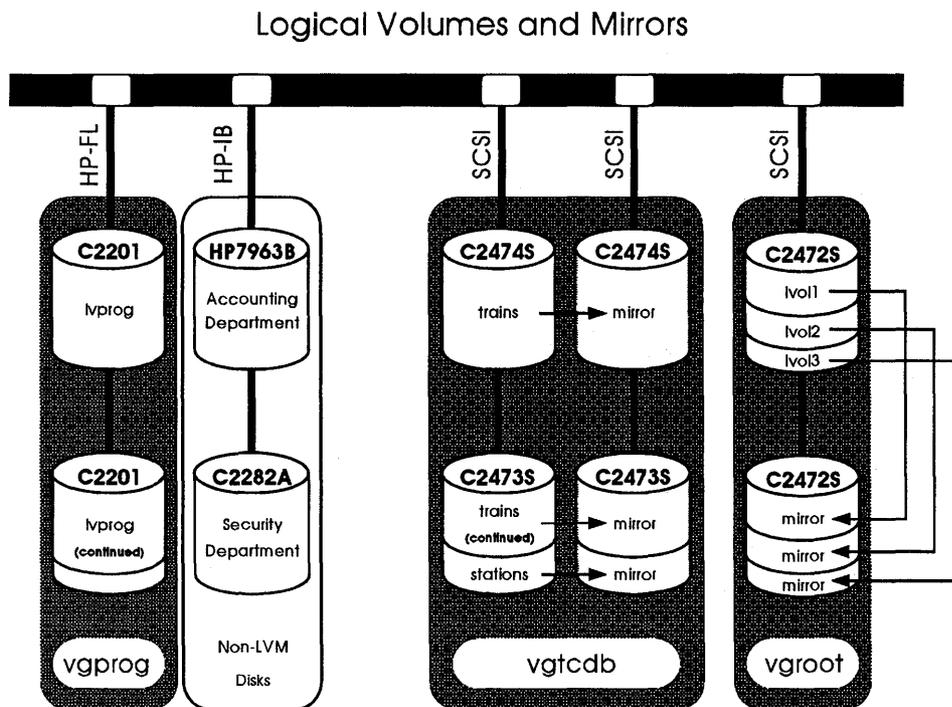
=====
Logical Volume Name      Size of LV      Used for:
=====
VG Name /dev/vgroot
LV Name /dev/vgroot/lvol1 127 Mb      Root file system (/)
LV Name /dev/vgroot/lvol2 205 Mb      Swap Space
LV Name /dev/vgroot/lvol3 90 Mb       /tmp

VG Name /dev/vgprog
LV Name /dev/vgprog/lvprog 1340 Mb     /programmers

VG Name /dev/vgtcdb
LV Name /dev/vgtcdb/trains 1704 Mb     Raw I/O Train Control
LV Name /dev/vgtcdb/stations 124 Mb     Raw I/O Station Control

```

Here is that information in a pictorial form, along with mirroring information that we gathered during part 4.



Computer Type: HP9000 Model 857
 System Name: Erall1
 HP-UX Revision: A.09.00

Figure 8-4. The System's Volume Groups and Logical Volumes

Although space in this manual does not permit us to print every `lvdisplay` output (generated during part 4 of this procedure), you should have hardcopies of these in your folder of information.

EXAMPLE:

```
/etc/lvdisplay -v /dev/vgtdcb/trains
--- Logical volumes ---
LV Name           /dev/vgtdcb/trains
VG Name           /dev/vgtdcb
LV Permission      read/write
LV Status          available/syncd
Mirror copies      1
Consistency Recovery MWC
Schedule           parallel
LV Size (Mbytes)   1704
Current LE         426
Allocated PE       852
Bad block          off
Allocation         strict

--- Distribution of logical volume ---
PV Name           LE on PV  PE on PV
/dev/dsk/c0d0s2   426       426
/dev/dsk/c9d0s2   426       426

--- Logical extents ---
LE  PV1                PE1  Status 1  PV2                PE2  Status 2
0000 /dev/dsk/c0d0s2   0000 current /dev/dsk/c9d0s2   0000 current
0001 /dev/dsk/c0d0s2   0001 current /dev/dsk/c9d0s2   0001 current
0002 /dev/dsk/c0d0s2   0002 current /dev/dsk/c9d0s2   0002 current
      :
      :
0424 /dev/dsk/c0d0s2   0424 current /dev/dsk/c9d0s2   0424 current
0425 /dev/dsk/c0d0s2   0425 current /dev/dsk/c9d0s2   0425 current
```

Recovering an lvmtab File - Using vgscan

The file `/etc/lvmtab` lists all volume groups and the physical volumes that are associated with each volume group. A number of LVM commands use the information in `/etc/lvmtab`, so it is important that `lvmtab` is present and up to date. During boot, for example, the disks and volume groups are activated based on the contents of the `/etc/lvmtab` file. If the `/etc/lvmtab` file has been destroyed or corrupted, the proper activation of volume groups and disks cannot occur.

Although it is sometimes hard to know when `/etc/lvmtab` has disappeared or has been corrupted, the following things might indicate that this has happened:

- Messages indicating that a volume group or disk that you know is present cannot be found
- LVM reports that it is creating an `lvmtab` file when you use a `display` command (such as `/etc/vgdisplay`), when an `lvmtab` file has always existed.
- Messages indicating that `/etc/lvmtab` is not present or has been corrupted.

You can use the `/etc/vgscan` command to recreate the `/etc/lvmtab` file. When the command is run, it scans each disk in the system, looking for logical volume and volume group information. It also scans the `/dev` directory, looking for matches between the volume group device files, the logical volume device files, and the disks.

Here are some important things to consider when you run `vgscan`:

- Run `vgscan` as soon as you realize the `/etc/lvmtab` has been damaged or destroyed. Remove a corrupt `/etc/lvmtab` file before recreating it with `vgscan`.
- If you have discovered that your `/etc/lvmtab` has been lost or damaged, run `vgscan` *before* you reboot your system.
- When you run `vgscan`, make sure all disks on your system are online. `vgscan` scans the disks that it can read for LVM configuration information. This is not an absolutely critical item, unless the disks that are off line will be needed to establish a quorum in a volume group. If a disk is not online when you run `vgscan`, its information will not be included in the new `lvmtab` file. When you return the disk to the volume group, the LVM commands that you use to do so will update `/etc/lvmtab` for you.

- Run `vgscan` with the `-p` and `-v` options first. The `-p` option allows you to preview how the new `/etc/lvmtab` file will be constructed, but does not create it. The `-v` option lets you view any message `vgscan` generates during the preview. Previewing what `vgscan` will do, allows you to look for missing disks or problems in your `/dev` directory before creating the `lvmtab` file.

Once you are confident that `/etc/lvmtab` will be built with the correct disks and volume groups, you can run `vgscan` without using the preview option, which will build the `lvmtab` file for you.

If `vgscan` finds disks that it cannot match with a volume group, or finds volume groups for which disks cannot be found, it instructs you to run `vgimport` to record the correct volume group and disk information in the `/etc/lvmtab` file. (See “Reconfiguring Disks in a Volume Group (Adjusting `/etc/lvmtab`)” for examples and a discussion about running `vgimport`.)

Reconfiguring Disks in a Volume Group (Adjusting `/etc/lvmtab`)

There are occasions when you might need to:

- Move the disks in a volume group to different hardware locations on a system.
- Move entire volume groups of LVM disks from one system to another.

When you do either of the above tasks, the LVM configuration file, `/etc/lvmtab`, must change to reflect the new hardware locations and device files for the disks.

The file `/etc/lvmtab` maintains information about all LVM disks in the system on a volume group basis. You cannot edit this file directly, since it is not a text file. Instead, you can use the `/etc/vgexport` and the `/etc/vgimport` commands to reconfigure the volume groups and record the configuration changes in the `/etc/lvmtab` file.

Moving a Volume Group's Disks

The procedure for moving the disks in a volume group to different hardware locations or different systems is illustrated in the following example.

Importing and Exporting Volume Groups

Suppose you want to move the three disks in the volume group `/dev/vg_planning` to another Series 800 computer.

1. Make the volume group unavailable.

```
vgchange -a n /dev/vg_planning
```

2. Use `vgexport(1M)` to remove the volume group information from the `/etc/lvmtab` file. You can first preview the actions of `vgexport` with the `-p` option.

```
vgexport -p -v -m plan_map vg_planning
```

With the `-m` option, you can specify a **map file** to hold the information that is removed from the `/etc/lvmtab` file. This file is important because it contains the names of all logical volumes in the volume group.

Later, you can use the mapfile when you set up the volume group on the new system.

If the preview is satisfactory, run the command without `-p`.

```
vgexport -v -m plan_map vg_planning
```

When `vgexport` runs, it removes the device files for the disks in the volume group, removes the `vg_planning` information from the `/etc/lvmtab` file, and creates the mapfile `plan_map`.

Once the `/etc/lvmtab` file no longer has the `vg_planning` volume group configured, you can shut down the system, disconnect the disks, and reinstall the disks on the new system. Transfer the file `plan_map` to the `/` directory on the new system.

3. Add the disks to the new system.

Once you have the disks installed on the new system, note their new LU numbers so you can refer to the device files created for them. Suppose, for our example, the new LU numbers are 6, 7, and 8.

4. On the new system, create a new volume group directory and `group` file. These steps are required for an imported volume group just as they are required for creating a new volume group (see “Creating a Volume Group” earlier in this chapter).

```
cd /  
mkdir dev/vg_planning  
cd dev/vg_planning
```

When you create the group file, specify a minor number that reflects the volume group number. (Volume group numbering starts at 0; the volume group number for the fifth volume group, for example, is 04.)

```
mknod /dev/vg_planning/group c 64 0x040000
```

5. Now, issue the `vgimport` command. To preview, use the `-p` option.

```
vgimport -p -v -m plan_map /dev/vg_planning /dev/dsk/c6d0s2 /dev/dsk/c7d0s2 /dev/dsk/c8d0s2
```

To actually import the volume group, re-issue the command omitting the `-p`.

6. Finally, activate the newly imported volume group:

```
vgchange -a y /dev/vg_planning
```

Where to go for more information

- For information on using the Logical Volume Manager, see “Managing Logical Volumes” in the *System Administration Tasks* manual.
- 8 ■ For more conceptual information about the Logical Volume Manager and its implementation, see Chapter 9, “Logical Volume Manager” in the manual *How HP-UX Works: Concepts for the System Administrator*.

Problems with Terminals

There are a number of terminal related problems that can occur. Many of these result in a terminal that appears not to communicate with the computer. Other problems cause “garbage” to appear on the screen (either instead of the data you expected or intermixed with your data).

This chapter primarily addresses problems with alpha-numeric display terminals; however, many of the steps discussed here can also be applied to problems with terminal emulators such as HP’s AdvanceLink (running on a Vectra PC) or X-Windows terminal processes (such as hpterm and xterm). We will look primarily at problems with unresponsive terminals. There are more of these than the other types of problems. We’ll look at some of the other types of problems at the end of this chapter.

Unresponsive Terminals

There are many things that can cause a terminal not to respond (no characters are displayed except, perhaps, those which are displayed by the terminal’s local echo setting). Here is a procedure you can use to find many of them.

Step 1: Check the status of the system

Is the system still up?

If not, you've probably found your problem. You will need to reboot the system.

Is the system in single-user mode?

If so, the only active terminal will be the system console. Other terminals will not respond. You will need to switch to a multi-user state (see the manual reference page for *init(1m)* for more information on changing run states.

Note

To check what run state your system is in (from a working terminal) type:

```
who -r
```

The output will look something like:

```
.          system boot  Feb 10 07:10    2    0    S
```

The current state of the machine is in the field immediately to the right of the time (third field from the right). For complete information on each of the fields, consult the *who(1)* manual reference page.

Step 2: Check to see if an editor is running on the terminal

This is best done from another terminal. Issue the command:

```
ps -ef
```

Look in the column marked TTY for *all* processes associated with the terminal you are having problems with. For each entry, check in the column marked COMMAND to see if the process represented by that entry is an editor.

If you find that an editor *is* running at the terminal, it is probably in a text-entry mode. You will need to save the work and exit the editor. For directions on how to do this, consult the manual reference page for the appropriate editor.

Caution If you are not sure of the status of the work being edited, *DO NOT* simply save the file and exit. You will overwrite the previous contents of the file with unknown text. Save the work in progress to a temporary file so that both the original and edited versions of the file are accessible.

Step 3: Enter `ctrl-q` at the terminal keyboard

Terminals frequently use the XON/XOFF protocol to start and stop output to them. If output to the terminal was stopped because an XOFF signal (`ctrl-s`) was sent from the terminal to the computer, it can be restarted by sending the computer an XON signal (type `ctrl-q` from the problem terminal's keyboard). Sending the XON signal does not harm anything even if no XOFF signal was previously sent.

If the problem is an application program that's looping or not functioning properly, try pressing the `break` key and then try `ctrl-C` to see if you can get a shell prompt back (`ctrl-C` is the default interrupt character; you might use a different one). If you need to find out what the interrupt character for the affected terminal is, go to a working terminal and enter the command:

```
stty < /dev/{device file name for the problem terminal}
```

Caution The `stty` command, above, should only be used with device file names for *currently active* terminal device files (use the `who(1)` command to see which device files are active). If you attempt to execute `stty` with a non-active device file, you will hang the terminal you entered the command from.

Step 4: Reset the terminal

The terminal itself may be stuck in an unusable state. Try resetting it. Consult your terminal owner's manual for information on how to do this. Powering the terminal off, waiting for a few seconds and powering it back on will reset the terminal (but there is probably an easier and better way to do it).

Step 5: Check the terminal configuration

The terminal might not be configured correctly. You should check the following:

- Is the terminal in **Remote *** mode? *It should be.*
- Is **Block *** mode turned ON? *It shouldn't be*
- Is **Line *** mode turned ON? *It shouldn't be*
- Is **Modify *** mode turned ON? *It shouldn't be*

Step 6: Check the physical connection

Check to make sure that:

- All cables are firmly attached and in their proper locations.
- All interface cards are firmly seated in their slots.
- The power cord to the terminal is firmly connected.
- The power switch is turned on.

Step 7: Kill processes associated with the problem terminal

Caution Use *extreme caution* when killing processes. The processes will be immediately and unconditionally terminated, so be sure you are not killing a valid process that just happens to be taking a long time to complete. Be sure not to make typos when entering the PID numbers for the `kill` command. You could accidentally kill the wrong process.

If you have another terminal that is still working, go to that terminal and login (you will need to be superuser). Execute the command:

```
ps -ef
```

The output will look similar to this:

UID	PID	PPID	C	STIME	TTY	TIME	COMMAND
root	95	1	0	Jul 20	?	0:00	/etc/getty -h ttyd1p0 9600
root	94	1	0	Jul 20	tty0p5	0:00	/etc/getty -h tty0p5 9600
root	22095	1	0	13:29:17	?	0:00	/etc/getty -h ttyd2p1 9600
root	22977	1	0	14:42:28	?	0:00	/etc/getty -h ttyd2p0 9600
root	14517	1	0	Jul 21	ttyd1p4	0:01	-csh [csh]
root	107	1	0	Jul 20	?	0:00	/etc/getty -h ttyd3p0 9600
stevem	20133	1	0	11:20:24	ttyd2p5	0:00	-csh [csh]
root	22147	1	0	13:33:45	?	0:00	/etc/getty -h ttyd2p3 9600
judyl	1159	1158	0	Jul 20	ttyp3	0:03	-csh [csh]
stevem	21234	20133	0	12:22:05	ttyd2p5	0:01	rlogin remote
petern	23367	23366	0	15:41:29	ttyp1	0:02	-csh [csh]
stevem	21235	21234	0	12:22:12	ttyd2p5	0:04	rlogin remote

Look in the column marked TTY for those processes that are associated with the terminal you are having problems with. Look at the column marked PID for those entries (these are the process IDs for the processes associated with that terminal). Execute the following command, listing each process ID associated with the problem terminal:

```
kill -9 {process-id} [{process-id} ... ]
```

If, in the example above, we wanted to kill the processes associated with terminal *ttyd2p5*, we would execute the command:

```
kill -9 21235 21234 20133
```

This should kill all processes associated with that terminal. The *init* process will then respawn a *getty* process for that terminal (if it has been set up to do that, in the */etc/inittab* file) and you should once again be able to login.

Step 8: Attempt to re-login to the previously hung port

Attempt to re-login to the previously hung terminal. If you are successful, you've fixed the problem. If not, continue to the next step.

Step 9: Use *cat* to send an ASCII file to the hung terminal's device file

HP-UX communicates with peripherals through device files. These special files are typically located in the */dev* directory, and are used by HP-UX to determine which driver should be used to talk to the device (by referencing the **major number**) and to determine the address and certain characteristics of the device HP-UX is communicating with (by referencing the **minor number**).

Try using the */bin/cat* command to send an ASCII file (such as */etc/motd* or */etc/issue*) to the device file associated with the problem terminal. For example, if your problem terminal is associated with the device file *ttyd1p4*:

```
cat /etc/motd > /dev/ttyd1p4
```

You should expect to see the contents of the file */etc/motd* displayed on the terminal associated with the device file */dev/ttyd1p4*. If you do not, continue to the next step.

Step 10: Check the parameters of the device file for the problem terminal.

Device files have access permissions associated with them, just as other files do. The file's access permissions must be set so that you have access to the file. If you set the files permissions mode to 622 (crw—w—w-), you should be safe.

If the file's permissions are set to allow write access and the file isn't displayed on the terminal, check the major and minor numbers of the device file. You can list them with the `ll` command. Consult *How HP-UX Works: Concepts for the System Administrator* for information about the format of minor numbers and the manual, *Installing Peripherals* for information on what the major number should be. If your computer is a Series 800 computer, you can use the `lssf` command to interpret the major and minor numbers for you and display the results.

Step 11: Other things to check

Make sure your inittab entries are active (telinit -q)

If you are just adding this terminal and have made a new entry in the `/etc/inittab` file by editing it, remember that this doesn't automatically make your new entry active. To do that you need to enter the command:

```
telinit -q
```

This tells the `init` process to scan the `/etc/inittab` file to update the information in its internal tables.

Check for functioning hardware

Now is the time to check the functionality of your hardware. To do this, check the following items:

- If your terminal has a self-test feature, activate it. If not, power the terminal off, wait several seconds, and power the terminal back on. This will test (at least to some degree) your terminal hardware.
- An alternate and perhaps better method to test the terminal hardware is to swap the suspect terminal with a known good one. This will allow for problems within the terminal that are *not* caught by the terminal selftest.

9

Note

Be sure to swap only the terminal (along with its keyboard and mouse); you want the known good terminal at the end of the SAME cable that the suspect terminal was plugged into). Also, plug the suspect terminal (with its keyboard and mouse) into the same cable that the known good terminal was plugged into and see if it functions there.

- If the known good terminal doesn't function on the suspect terminal's cable and the suspect terminal is working fine in its new location, you can be confident that the terminal itself is functioning properly and the problem is elsewhere.
 - The next thing to check is the cable connecting the terminal to the computer. Swap the suspect cable with a known good one.
-

Note

Since you know the terminal at the end of each cable is working, you only have to swap the ends of the cables where they connect to the computer. If the problem remains with the terminal it was associated with prior to the cable swap, you probably have a broken or miswired cable. If the problem transfers to the other terminal (and the previously bad terminal/cable combination works in its new location), then the problem is most likely with your MUX, port, or interface card.

Other Terminal Problems

The other type of problem you're likely to run into with terminals is that of garbage on the screen. Garbage on the screen comes in two types: garbage intermixed with valid data characters and complete garbage.

What to check for when garbage is mixed with valid data

The following is a list of possible reasons for garbage characters intermixed with your valid data:

- Noise on the data line:
 - RS-232 Cable too long (maximum recommended length is 50 feet)
 - Data cable near electrically noisy equipment (motors, etc.)
 - Partially shorted or broken wires within the cable
 - Noisy connection (if using phone lines)
- Hardware problem with a modem, interface card, or the terminal itself
- The program performing I/O could be sending the garbage
- The `Display Functns*` feature of your terminal is enabled (which displays characters that would not normally print)

What to check for when everything printed is garbage

One of the most common reasons for total garbage on the screen (and certainly the *first* thing you should check) is a Baud-rate mismatch. If your terminal's speed setting is different than that of the line (as set with the `stty` command), you will get garbage on your screen (if anything at all).

If you have not yet logged in, try pressing the `break` key. This tells `getty` to try the next entry in the `/etc/gettydefs` file. The `gettydefs` file can be set up so that, as `getty` tries various entries, it will also be trying various speed settings (this is usually how it's set up). `getty` will then try various speeds (with each press of the `break` key. When the correct speed is matched, you will get a login prompt that is readable.

Here is a list of other possible reasons for total garbage on your screen.

- The shell environment variable called “TERM” isn’t set to a value appropriate to your terminal. If you have an HP terminal, try setting the value of “TERM” to “hp” (lower case) using your shell’s `set` command.
- A running process is producing garbage output
- A miswired cable
- Excessive noise on the data line
- A hardware failure (bad interface card, modem, MUX, etc.)

Where to go for more information

- For more information about the `inittab` file and system run levels, refer to *How HP-UX Works: Concepts for the System Administrator*.



System Panics

System Panics: What They Are And Why They Happen

The term *panic* is, by definition, frightening! To see a message displayed on your system console that HP-UX has panicked can be alarming. But it is not necessary to panic when your system does. In HP-UX terms, a **panic** simply means that HP-UX ran into a condition that it did not know how to respond to, so it halted your computer.

System panics are rare and not always the result of a catastrophe. They sometimes occur on boot up if your system was previously not shut down properly. Sometimes they occur as the result of a hardware failure. In a clustered environment, a diskless client node will panic if too much time has elapsed since its last communication with its server. This could be the result of nothing more than a LAN cable that has been disconnected for too long.

Recovering from a system panic can be as simple as rebooting your system. If you have an up-to-date set of file system backup tapes, the *worst case* scenario would involve reinstalling HP-UX and restoring any files that were lost or corrupted (if this situation was caused by a hardware failure such as a disk head crash, you will, of course, have to have the hardware fixed before you can perform the reinstallation).

Note

It is important to maintain an up-to-date backup of the files on your system so that, in the event of a disk head crash or similar situation, you can recover your data. How frequently you update these backups depends on how much data you can afford to lose. For information on how to back up data, refer to *System Administration Tasks*.

What to do when your system panics

When HP-UX panics, it will display a “panic message” on the system console. When this happens, take the following steps.

Step 1: Record the *panic message* displayed on the system console

Write down the message that is displayed on the system console in case you need it later.

Step 2: Categorize the panic message

The panic message will tell you why HP-UX panicked. Sometimes panic messages refer to internal structures of HP-UX (or its file systems) and the cause might not be obvious. Generally, the problem is in one of the following areas, and wording of the message should allow you to classify it into one of them:

Category	Proceed to Step #
Hardware Failure	Step 3a
File system Problem (corrupted?)	Step 3b
LAN communication Problem	Step 3c
LVM-related Problem	Step 3d
None of the above	Step 3e

Step 3a: Hardware Failure Recovery

If the panic message indicated a hardware failure, the text or context of the message should indicate what piece of hardware failed.

If the hardware failure appears to be associated with a peripheral, check to be sure that its cables are tightly connected to their proper locations and that the device is powered on and in an “online” status. If there is an error indicated on the device’s display:

1. record the error message or display in your log book
2. turn the device off
3. if the device is a disk drive, wait for it to stop spinning
4. turn the device back on

If the problem reappears on the device or if the hardware failure appears to be associated with an interface card or an internal component of the System Processing Unit, it might be necessary to have the problem fixed by Hewlett-Packard or whoever performs your hardware maintenance.

Proceed to Step 4 (rebooting your system).

Step 3b: File system problem recovery:

If the panic message indicates a problem with one of your file systems, you will need to run the file system checker *fsck(1m)* to check and correct the problem(s). This is normally done automatically at boot time (from the */etc/rc* file) so you should proceed to step 4 (rebooting your system). Follow all directions that *fsck* gives you *especially if it is your root file system (the one with the “/” directory) that has the problem*. It is important to use the “-n” option to the *reboot(1m)* command if requested to do so by *fsck* during any subsequent reboot.

Step 3c: LAN communication problem

If the panic messages indicates a problem with LAN communication (such as when a diskless cluster client node is prevented from communication for too long), check all LAN cable connections to be sure of the following:

- All connectors are tightly fastened to the LAN cable and the media access units (MAU's). If you are using "thick LAN", make sure all vampire taps are tightly connected to their respective cables and that AUI cables are connected securely to the LAN interface cards (LANICs) in your computer.
- Your LAN is properly terminated. Each END of the LAN cable MUST have a *50 ohm* terminator on it. *Do NOT* connect a computer directly to the END of a LAN cable.

Proceed to step 4 (rebooting your system).

Step 3d: LVM-related Problem

If you reduce the size of a logical volume that contains a file system such that the logical volume is smaller than the file system within it, you will corrupt the file system. This will often manifest itself by causing your system to panic when you attempt to access a part of the truncated file system that is beyond the new boundary of the logical volume.

The problem might not show up immediately. This will occur when the truncated part of the file system is overwritten by something else (such as a new logical volume, or the extension of a logical volume in the same volume group as the truncated file system).

For more information on how to recover from this problem see "I Reduced the Size of a Logical Volume and Now My System Crashes!" in Chapter 8 of this manual.

For other LVM-related problems, see Chapter 8, "Logical Volume Manager (LVM) Problems".

Step 3e: Recovery from other situations

When you suspect the problem was something other than the above (or when you do not know where to classify it), proceed to step 4 (Rebooting your system). Many times, that's all that's required to recover from a system panic and it's certainly worth a try. In this case, it is *especially* important that you write down the *exact text* of the panic message, just in case you need it for future troubleshooting.

Step 4: Rebooting your system

Once you have checked for and corrected any problems from Step 3, you are ready to reboot your system. If your system has a "reset" switch or button, you can reboot your system using that. Otherwise, turn your computer off and then back on to initiate the boot up sequence.

You will probably notice a few differences in the boot up displays/activities as compared with your normal boot up sequence. Your computer might save a "core" file to disk. This core file is a "snapshot" of the previously running kernel at the time that it panicked. If it becomes necessary, this core file can be analyzed using special tools to determine more about what caused the panic.

Note

These core files are *big* and are saved to the directory `/tmp/syscore`. If you feel you need to save these files for future analysis (something that isn't usually required), it is best to save them to tape and remove them from your file system in order to free up space. If you *know* why your system panicked, you can delete the core files; it is unnecessary to keep them. The core files are used in rare circumstances to diagnose hard-to-find causes of system panics.

If the reason your system panicked was because of a corrupted file system, `fsck` will report the errors and any corrections it makes. If `fsck` terminates and requests to be run manually, refer to chapter 6 (File System Problems) for further instructions. If the problems were associated with your root file system, `fsck` will ask you to reboot your system when it's finished. When you do this, use the command:

```
reboot -n
```

The `-n` option tells `reboot` *not* to sync the file system before rebooting. Since `fsck` has made all the corrections on disk, you do not want to undo the changes by writing over them with the still corrupt memory buffers.

If other problems occur during the boot process, refer to chapter 5 (System Boot-up Problems).

Step 5: Monitor the system closely for a while

If your system successfully boots, there is a good chance that you can resume normal operations. Many system panics are isolated events, unlikely to reoccur.

Check your applications to be sure that they are running properly and (for a day or so) monitor the system closely. For a short while, you might want to do backups more frequently until you are confident that the system is functioning properly.

Using the fsck Command

The file system consistency check(`/etc/fsck`) checks for and repairs inconsistencies in your file system.

You must have a thorough understanding of the file system before making any `fsck` decisions.

The `fsck` command should be performed:

- During bootup, if you did not have a clean shutdown (if you did not use `shutdown` or `reboot`).
- Run `fsck` before the system is taken to run-level 2. As shipped, your system should do this automatically if it detects an improper shutdown via the `bcheckrc` entry in `/etc/inittab`. An improper shutdown is one where you did not use the `shutdown` command described in the “System Shutdown” section in Chapter 3.
- Any time you suspect problems with the HP-UX file system.

The `fsck` program, when run on the root file system, *must* use the block device (for example, `/dev/rdisk/c0d0s0`).

The `fsck` program can be run in several different modes.

`-p` Preening mode

This option fixes many potential problems, but never removes data. When you preen the system, you are not running interactively. The `fsck` command determines what to do, and if it cannot deal with a situation, it terminates. For any inconsistencies preening mode fixes, it prints a message identifying the file system, and the corrective action taken. The preening option can fix the following inconsistencies:

- unreferenced inodes
- unreferenced pipes and fifos

A

- link counts in inodes too large
- missing blocks in the free list
- blocks in the free list also in files
- wrong counts in the superblock
- clean byte marked wrong

Other problems terminate `fsck -p` and prompt for manual execution of `fsck`.

`-P` Preening mode.

This option is used by `/etc/bcheckrc`. It operates in the same manner as the `-p` option except it ignores those file systems marked clean by commands such as `umount` and `reboot`.

`-y` Yes mode.

Using the `-y` option can be very dangerous. This option causes `fsck` to answer YES to all questions, which might remove data. *Do not use the `-y` option if you have important data on your file system unless you have first used the `-n` option and understand the potential damage.*

`-n` No mode

Using the `-n` option causes `fsck` to answer NO to all questions. This option never removes data, so it is safe. You can use the `-n` option anytime: in multiuser (though not recommended) or single-user mode, or in the background.

If you use `fsck` with the `-n` option in multiuser mode, you will probably come up with some inconsistencies due to file system action. However, you will not damage your system.

`de-` Interactive mode.

`fault`

The interactive mode allows you to choose whether to perform each action or not.

`-q` Quiet mode.

The `fsck` command prints only the messages that require a response.

The system should *always* be in a single-user state and quiescent (inactive and not being written on) with all file systems unmounted before executing the

A-2 Using the `fsck` Command

fsck command. The only exception is the root, which is always mounted. Use **fsck** in run-level **s** after the **shutdown** command is executed. Running **fsck** when there is file system activity can cause loss of data.

The **fsck** command should be executed using a character special device file, not a block special file, except for the root file system. Refer to the section “Adding and Removing Peripheral Devices” in Chapter 5 for a discussion on block and character devices, and for naming conventions for device files.

Only the System Administrator should run **fsck**. If this check discovers an inconsistency, corrective action must be taken.

Before running **fsck**, make sure that a directory called **/lost+found** exists on the file system you plan to examine. One **/lost+found** should be created for each file system when you installed HP-UX, or when you ran **newfs** or **mkfs**. The **fsck** command uses this directory to place any problem files or directories that it finds. After you run **fsck**, examine the files placed in **/lost+found** and move them back where they belong or remove them. You should clear the **/lost+found** directory before executing **fsck** again.

To place these files, follow this procedure:

1. Mount the file system.
2. Change to the **lost+found** directory (**cd /lost+found**).
3. Find out what type of file it is (executable, text, etc.) and who owns it by typing:

```
file *  
ll *
```

If the file is text, you can examine its contents by typing “**more filename**”, where **filename** is the name of the file.

4. If the file is executable, you can try one of two things:
 - a. If the file has an SCCS ID string, the **what** command will list it.
 - b. If the file does not have an SCCS ID string, use the **strings** command to print the literal strings from the file. The strings (such as error message strings) might help identify the owner.
5. From this information, determine where the file belongs, or who it belongs to, and move the file to the correct directory.

How fsck Handles Inconsistencies

The **fsck** command is a multi-pass file system check program. Each phase of the **fsck** program invokes a different file system pass. After the initial setup, **fsck** performs successive passes over each file system, checking blocks and sizes, path names, connectivity, reference counts, and the free block map (possibly rebuilding it), and doing some cleanup.

Refer to the beginning of this appendix for a description of the different modes in which you can run **fsck**.

When an inconsistency is detected while running interactively, **fsck** reports the error condition. If a response is required, **fsck** prints a prompt message and waits for a response. When preening, **fsck** will choose a response and note it on the screen. In this section, each error message and possible responses are presented.

The error conditions are organized by the phase of the **fsck** program in which they can occur. The error conditions that can occur in more than one phase are discussed in “Initialization Phase Errors” below.

Initialization Phase Errors

During the initialization phase and before the file system check is performed, tables have to be set up and certain files opened. This section lists error conditions resulting from command line options, memory requests, opening of files, status of files, file system size checks, and creation of the scratch file. All of the initialization errors are fatal if you are preening. See the *fsck(1m)* entry in the *HP-UX Reference* for further information.

“C” option?

The character represented by **C** is not a legal option for **fsck**. Legal options are **-b**, **-y**, **-n**, **-q**, **-P**, and **-p**. The **fsck** command terminates on this error condition. See the *fsck(1m)* entry in the *HP-UX Reference Manual* for further information.

cannot alloc NNN bytes for “XXX”

XXX is either **blockmap**, **freemap**, **statemap**, or **lncntp**. The **fsck** command’s request for memory failed. This should never happen; **fsck** terminates on this error condition. Contact your local HP Sales and Service Office for assistance.

Can't open checklist file:

The default file system checklist file (`/etc/checklist`) cannot be opened for reading. The `fsck` command terminates on this error condition. Check for the existence of the file and the access modes of the file.

Can't stat root

The `fsck` command's request for statistics about the root directory (`/`) failed. This should never happen; `fsck` terminates on this error condition. Contact your local HP Sales and Service Office for assistance.

Can't stat ...**Can't make sense out of ...**

The `fsck` command's request for statistics about the file system failed. When running manually, it ignores this file system and continues checking the next file system given. If this happens, check for the existence and the access modes of the file system.

"file_system_name" is not a block or character device; OK?

You have given `fsck` a regular file name by mistake. You should check the file type of the file system. Possible responses to the "OK" prompt are:

YES Ignore this error condition.

NO Ignore this file system and continue checking the next file system given.

Can't open ...

The file system listed cannot be opened for reading. When running manually, it ignores this file system and continues checking the next file system given. Check the access modes of the file system.

"file_system_name": (NO WRITE)

Either the `-n` flag was specified or `fsck`'s attempt to open the file system, "file_system_name", for writing failed. When running manually, all the diagnostics are printed, but no modifications are attempted to fix them.

Other messages:

A

MAGIC NUMBER WRONG

NGC OUT OF RANGE

CPG OUT OF RANGE

NCYL DOES NOT GIVE WITH NCG*CPG

SIZE PREPOSTEROUSLY LARGE

TRASHED VALUES IN SUPER BLOCK

will be followed by the message:

file_system: BAD SUPER BLOCK: *superblock_address* USE -b OPTION OF FSCK TO SPECIFY LOCATION OF AN ALTERNATE superblock TO SUPPLY NEEDED INFORMATION; SEE *fsck*(1m).

The superblock is corrupted. An alternative superblock must be used. See the discussion on alternative superblocks under the “Superblock Consistency” section in this appendix.

INTERNAL INCONSISTENCY: *message*

An internal problem occurred in *fsck*. *message* indicates the problem. This should never happen. Contact your local HP Sales and Service Office for assistance.

CANNOT SEEK: BLK *bn* (CONTINUE)?

The *fsck* command’s request for moving to the specified block number in the file system failed. This should never happen. Contact your local HP Sales and Service Office for further assistance. Possible responses to the “CONTINUE” prompt are:

YES Attempt to continue to run the file system check. Often, however, the problem will persist. This error condition will not allow a complete check of the file system. Run *fsck* a second time to recheck this file system.

NO Terminate the program.

CANNOT READ: BLK ... (CONTINUE)?

The **fsck** command's attempt to read a specified block number in the file system failed. This can happen when you interrupt **fsck** before it finishes. Contact your local HP Sales and Service Office for further assistance.

Possible responses to the "CONTINUE" prompt are:

YES Attempt to continue to run the file system check. Often, however, the problem will persist. This error condition will not allow a complete check of the file system. Run **fsck** a second time to recheck this file system.

NO Terminate the program.

CANNOT WRITE: BLK ... (CONTINUE)?

The **fsck** command's attempt to write a specified block number in the file system failed. The disk is probably physically write-protected . Remove write protection from the disk, and rerun **fsck**.

Possible responses to the "CONTINUE" prompt are:

YES Attempt to continue to run the file system check. Often, however, the problem will persist. This error condition prevents a complete check of the file system. Run **fsck** a second time to recheck this file system.

NO Terminate the program.

Phase 1 Errors: Check Blocks and Sizes

This phase concerns itself with the inode list. This section lists error conditions resulting from checking inode types, setting up the zero-link-count table, examining inode block numbers for bad or duplicate blocks, checking inode size, checking block count, and checking inode format. All errors in phase 1 are fatal if you are preening the file system, except for **INCORRECT BLOCK COUNT**, **BAD INDIRECT ADDRESS**, and **NON-ZERO READER/WRITER COUNT**.

CG ... : BAD MAGIC NUMBER

The magic number of cylinder group is wrong. This usually indicates that the cylinder group maps have been destroyed. When running manually, the cylinder group is marked as needing to be reconstructed.

NON-ZERO READER/WRITER COUNT(S) ON PIPE I= ... (CORRECT)?

The inode indicates that a process is reading or writing from or to the pipe. Possible responses to “CORRECT” prompt are:

YES Restart the number of readers and writers of this pipe to 0.

NO Ignore this error condition.

BAD DIRECT ADDRESS, SHOULD BE ZERO: inode.didb[n]= ... (CORRECT)?

The inode contains a direct disk block address for regions beyond the allocated size of the file. During the preening process, these entries are zeroed.

Possible responses to the “CORRECT” prompt:

YES Zero the entry.

NO Ignore this error condition. Later attempts by the operating system to extend the file into this region might cause a system crash.

UNKNOWN FILE TYPE I= ... (CLEAR)?

The mode word of the inode indicates that the inode is not a character special, block special, regular, network special, fifo, symbolic link, or directory inode.

Possible responses to the “CLEAR” prompt are:

YES Deallocate the inode by zeroing its contents. This always invokes the UNALLOCATED error condition in Phase 2 for each directory entry pointing to this inode.

NO Ignore this error condition.

LINK COUNT TABLE OVERFLOW (CONTINUE)?

An internal table for `fsck` containing allocated inodes with a link count of zero has no more room.

Possible responses to the “CONTINUE” prompt are:

YES Continue with the program. This error condition prevents a complete check of the file system. Run `fsck` a second time to recheck this file system. If another allocated inode with a zero link count is found, this error condition is repeated.

NO Terminate the program.

“block_number ”BAD I= ...

The inode represented by “I= ... ” contains the block number *block_number*. This block number is out of the range of the file system. This error condition might invoke the EXCESSIVE BAD BLKS error condition in phase 1 if this inode has too many block numbers outside the file system range. This error condition always invokes the BAD/DUP error condition in Phase 2 and Phase 4.

EXCESSIVE BAD BLKS I= ... (CONTINUE)?

There are more than 10 blocks with a block number out of the range of the file system associated with the inode.

Possible responses to the “CONTINUE” prompt are:

YES Ignore the rest of the blocks in this inode and continue checking with the next inode in the file system. This error condition will not allow a complete check of the file system. Run `fsck` a second time to recheck this file system.

NO Terminate the program.

block_number DUP I= ...

The inode contains block number, *block_number*, which is already claimed by another inode. This error condition might invoke the EXCESSIVE DUP BLKS error condition in phase 1 if this inode has too many block numbers claimed by other inodes. This error condition will always invoke Phase 1b and the BAD/DUP error condition in Phase 2 and Phase 4.

EXCESSIVE DUP BLKS I= ... (CONTINUE)?

There are more than 10 blocks claimed by other inodes.

Possible responses to the “CONTINUE” prompt are:

YES Ignore the rest of the blocks in this inode and continue checking with the next inode in the file system. This error condition prevents a complete check of the file system. Run **fsck** a second time to recheck this file system.

NO Terminate the program.

DUP TABLE OVERFLOW (CONTINUE)?

An internal table in **fsck** containing duplicate block numbers is full.

Possible responses to the “CONTINUE” prompt are:

YES |Continue with the program. This error condition prevents a complete check of the file system. Run **fsck** a second time to recheck this file system. If another duplicate block is found, this error condition repeats.

NO Terminate the program.

PARTIALLY ALLOCATED INODE I= ... (CLEAR)?

The bitmap of the file system is inconsistent with inode status.

Possible responses to the “CLEAR” prompt are:

YES Deallocate the inode by zeroing its contents.

NO Terminate the program.

INCORRECT BLOCK COUNT I= ... (CORRECT)?

The block count for the inode, *inode_number*, is *X* blocks, but should be *Y* blocks. When you are preening the count is corrected.

Possible responses to the “CORRECT” prompt are:

YES replace the block count of the inode with *Y*.

NO ignore this error condition.

BAD INDIRECT ADDRESS: IND BLOCK n[m] = val I= ... (CORRECT)?

An indirect address block, allocated in the inode indicated by “I= ...”, contains block address for regions beyond the allocated size of the file. When you are preening, these entries are zeroed.

Possible responses to the “CORRECT” prompt are:

YES Zero the entry.

NO Ignore this error condition. Later attempts by the operating system to extend the file into this region can cause a system crash.

Phase 1b: Rescan for More Dups

When a duplicate block is found in the file system, the file system is rescanned to find the inode which previously claimed that block. This section lists the error condition when the duplicate block is found.

block_number DUP I= ...

The inode contains the block number, *block_number*, which is already claimed by another inode. This error condition will always invoke the BAD/DUP error condition in Phase 2. You can determine which inodes have overlapping blocks by examining this error condition and the DUP error condition in Phase 1.

Phase 2: Check Path Names

This phase concerns itself with removing directory entries pointing to error conditioned inodes from Phase 1 and Phase 1b. This section lists error conditions resulting from root inode mode and status, directory inode pointers in range, and directory entries pointing to bad inodes. All errors in this phase are fatal if you are preening your file system.

ROOT INODE UNALLOCATED. TERMINATING

A

The root inode (inode number 2) has no allocated mode bits. This should never happen. The program will terminate. Contact your local HP Sales and Service Office for further assistance.

NAME TOO LONG

The path name shown is too long. This is usually indicative of loops in the file system name space. This can occur if the super user has made circular links to directories. The offending links must be removed.

ROOT INODE NOT DIRECTORY (FIX)?

The root inode (inode number 2) is not directory inode type.

Possible responses to the "FIX" prompt are:

YES Change the root inode's type to be a directory. If the root inode's data blocks are not directory blocks, many error conditions are produced.

NO Terminate the program.

DUPS/BAD IN ROOT INODE (CONTINUE)?

Phase 1 or Phase 1b found duplicate blocks or bad blocks in the root inode (inode number 2) for the file system.

Possible responses to the "CONTINUE" prompt are:

YES Ignore the DUPS/BAD error condition in the root inode and attempt to continue to run the file system check. If the root inode is not correct, this can result in a large number of other error conditions.

NO Terminate the program.

I OUT OF RANGE I= ... DIR=|NAME (REMOVE)?

NAME has an inode number *I*, which is greater than the end of the inode list.

Possible responses to the "REMOVE" prompt are:

YES The directory entry *NAME* is removed.

NO Ignore this error condition.

UNALLOCATED I= ... (REMOVE)?

Two possible error messages begin like this. One is for directory entries and the other is for files. A directory has an entry for a directory or file, but the inode I for the directory or file is not allocated. The owner, mode, size, modify time, and directory or file name are printed.

Possible responses to the "REMOVE" prompt are:

YES The directory entry is removed.

NO Ignore this error condition.

DUP/BAD I= ... (REMOVE)?

There are two possible error messages that start like this. One is for directory entries and one is for files. Phase 1 or Phase 1b found duplicate blocks or bad blocks associated with the directory or file having inode I. The owner, mode, size, modify time, and directory are printed.

Generally, the inode with the earliest modify time is incorrect, and should be cleared.

Possible responses to the "REMOVE" prompt are:

YES The directory entry is removed.

NO Ignore this error condition.

ZERO LENGTH DIRECTORY I= ... (REMOVE)?

The directory entry's size is zero. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the "REMOVE" prompt are:

YES the directory entry is removed. This will always invoke the BAD/DUP error condition in phase 4.

NO ignore this error condition.

DIRECTORY TOO SHORT I= ... (FIX)?

The directory entry's size is less than the minimum size for a directory. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the "FIX" prompt are:

YES increase the size of the directory to the minimum directory size.

NO ignore this error condition.

DIRECTORY CORRUPTED I= ... (FIX)?

A directory entry has an inconsistent internal state. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the "FIX" prompt are:

YES Throw away all entries up to the next directory boundary. This drastic action can throw away directory entries, and should be taken only after other recovery efforts have failed.

NO Skip to the next directory boundary and resume reading, but do not modify the directory.

BAD INODE NUMBER FOR '.' I= ... (FIX)?

The directory entry doesn't have an inode number for '.', which is equal to the inode number. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the "FIX" prompt are:

YES change the inode number for '.' to be equal to the inode number given after 'I='.

NO leave the inode number for '.' unchanged

MISSING '.' I= ... (FIX)?

The directory doesn't have its first directory entry allocated. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the FIX prompt are:

YES make an entry for '.' with inode number equal to the inode number given after 'I='.

NO leave the directory unchanged.

MISSING '.' I= ...

CANNOT FIX, FIRST ENTRY IN DIRECTORY CONTAINS ...

The directory has, as its first entry, the file name given. The **fsck** command cannot resolve this problem. The file system should be mounted and the offending entry moved elsewhere. To do this, exit the **fsck** program, mount the file system (you can force a mount by using `mount file_system -f`, find the file name, and move it to a different directory. The file system should then be unmounted and **fsck** should be run. The owner, mode, size, modify time, and directory name are printed.

MISSING '.' I= ...

CANNOT FIX, INSUFFICIENT SPACE TO ADD '.'

The directory does not have '.' as its first entry. The **fsck** command cannot resolve this problem. If this happens, contact your local HP Sales and Service office. The owner, mode, size, modify time, and directory name are printed.

EXTRA '.' ENTRY I= ... (FIX)?

The directory has more than one entry for '.'. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the "FIX" prompt are:

YES remove the extra entry for '.'.

NO leave the directory unchanged.

BAD INODE NUMBER FOR '..' I= ... (FIX)?

The directory's inode number for '..' does not equal the parent of the inode number *I*. The owner, mode, size, modify time, and directory name are printed.

Possible responses to the "FIX" prompt are:

YES change the inode number for '..' to be equal to the parent of the inode given *I*.

NO leave the inode number for '..' unchanged.

MISSING '..' I= ... (FIX)?

The directory doesn't have its second directory entry allocated.

Possible responses to the "FIX" prompt are:

YES make an entry for '..' with inode number equal to the parent of the inode number given *I*.

NO leave the directory unchanged.

MISSING '..' I= ...

CANNOT FIX, SECOND ENTRY IN DIRECTORY CONTAINS ...

The directory has, as its second entry, the file name given. The `fsck` command cannot resolve this problem. The file system should be mounted and the offending entry moved elsewhere. To do this, exit the `fsck` program, mount the file system (you can force a mount by using `mount file_system -f`, find the file name, and move the file to a different directory. The file system should then be unmounted and `fsck` should be run again.

MISSING '..' I= ...

CANNOT FIX, INSUFFICIENT SPACE TO ADD '..'

The directory does not have '..' as its second entry. The `fsck` command cannot resolve this problem. If this happens, contact your local HP Sales and Service office.

EXTRA '..' ENTRY I= ... (FIX)?

The directory has more than one entry for '..'.

Possible responses to the "FIX" prompt are:

YES remove the extra entry for '..'.

NO leave the directory unchanged.

UNUSED SPACE BETWEEN '.' AND '..' I= ... (FIX)?

There is enough empty space between '.' and '..' in this directory for a new entry to be allocated between the two entries. A new entry between '.' and '..' would violate the requirement that these two entries be the first two in a directory. This condition will typically occur when a file system has been incorrectly or incompletely converted to allow long filenames.

If you are preening, the directory is fixed.

Possible responses to the "FIX" prompt are:

YES copy the '..' next to the '.' to remove the gap between the two entries

NO leave the directory unchanged

Phase 3: Check Connectivity

This phase concerns itself with the directory connectivity seen in Phase 2. This section lists error conditions resulting from unreferenced directories, and missing or full `/lost+found` directories.

UNREF DIR I= ... (RECONNECT)?

The directory inode *I* was not connected to a directory entry when the file system was traversed. The owner, mode, size, and modify time of the directory inode are printed. If you are preening, the directory is reconnected if its size is non-zero, otherwise it is cleared.

Possible responses to the "RECONNECT" prompt are:

YES Reconnect the directory inode to the file system in the directory for lost files (`/lost+found`). This might invoke the "lost+found" error condition if there are problems connecting the directory inode to

A

`/lost+found`. This might also invoke the “CONNECTED” error condition in phase 3 if the link was successful.

NO Ignore this error condition. This error will always invoke the “UNREF” error condition in Phase 4.

SORRY, NO `lost+found` DIRECTORY

There is no `/lost+found` directory in the root directory of the file system. The `fsck` command ignores the request to link a directory in `/lost+found`. This will always invoke the UNREF error condition in Phase 4. Check access modes of `/lost+found`. This error is fatal if you are preening the system.

SORRY. NO SPACE IN `lost+found` DIRECTORY

There is no space to add another entry to the `/lost+found` directory in the root directory of the file system. The `fsck` command ignores the request to link a directory in `/lost+found`. This will always invoke the “UNREF” error condition in Phase 4. Clean out unnecessary entries in `/lost+found` or make `/lost+found` larger and try again. This error is fatal if you are preening the system.

DIR I= ... PARENT WAS I= ...

This is an advisory message indicating a directory inode was successfully connected to the `/lost+found` directory. The parent inode of the directory inode is replaced by the inode number of the `/lost+found` directory.

Phase 4: Check Reference Counts

This phase concerns itself with the link count information seen in Phase 2 and Phase 3. This section lists error conditions resulting from unreferenced files, missing or full `/lost+found` directories, incorrect link counts for files, directories, or special files, unreferenced files and directories, bad and duplicate blocks in files and directories, and incorrect total free-inode counts. All errors in this phase are correctable if you are preening your file system unless you run out of space in `/lost+found`.

UNREF FILE I= ... (RECONNECT)?

The inode *I* was not connected to a directory entry when the file system was traversed. The owner, mode, size, and modify time of the inode are printed. If

you are preening, the file is cleared if either its size, or its link count is zero. Otherwise it is reconnected.

Possible responses to the “RECONNECT” prompt are:

- YES* Reconnect the inode to the file system in the directory for lost files (/lost+found). This might invoke the “lost+found” error condition if there are problems connecting the inode to /lost+found.
- NO* Ignore this error condition. This will always invoke the “CLEAR” error condition in phase 4.

(CLEAR)?

The inode mentioned in the immediately previous error condition cannot be reconnected. If you are preening, this error cannot occur, since lack of space to reconnect files is a fatal error.

Possible responses to the “CLEAR” prompt are:

- YES* deallocate the inode mentioned in the previous error condition by zeroing its contents.
- NO* Ignore this error condition.

SORRY. NO lost+found DIRECTORY

There is no /lost+found directory in the root directory of the file system. The **fsck** command ignores the request to link a file in /lost+found. This will always invoke the “CLEAR” error condition in phase 4. Check access modes of /lost+found. If you are preening the file system, this error is fatal.

SORRY. NO SPACE IN lost+found DIRECTORY

There is no space to add another entry to the /lost+found directory in the root directory of the file system. The **fsck** command ignores the request to link a file in /lost+found. This always invokes the “CLEAR” error condition in phase 4. Check the size and contents of /lost+found. This error is fatal if you are preening the file system.

LINK COUNT ... (ADJUST)?

The link count for the file, directory, or inode is one link count *COUNT*= but should be a different link count. The owner, mode, size, and modify time are printed. If you are preening, the link count is adjusted.

Possible responses to the “ADJUST” prompt are:

YES Replace the link count of the file in the inode with what the number should be.

NO Ignore this error condition.

UNREF ... (CLEAR)?

The file or directory with the inode number, *I*, was not connected to a directory entry when the file system was traversed. The owner, mode, size, and modify time of the inode are printed. If you are preening, the inode is cleared, since this is a file that was not connected because its size or link count was zero.

Possible responses to the “CLEAR” prompt are:

YES deallocate inode by zeroing its contents.

NO Ignore this error condition.

BAD/DUP ... (CLEAR)?

Phase 1 or Phase 1b found duplicate blocks or bad blocks associated with the file or directory inode given in *I*. The owner, mode, size, and modify time of the inode are printed. This error does not occur if you are preening, since it would have caused a fatal error earlier.

Possible responses to the “CLEAR” prompt are:

YES deallocate the inode by zeroing its contents.

NO Ignore this error condition.

Often deleting only one of the files containing DUPS will cure the problem. The **fsck** command should be rerun to confirm that the problem was fixed. A “NO” means that **fsck** must be rerun to finish cleaning up the file system.

FREE INODE COUNT WRONG IN SUPERBLK (FIX)?

The actual count of the free inodes does not match the count in the superblock of the file system. If you are preening, the count is fixed.

Possible responses to the "FIX" prompt are:

YES Replace the count in the superblock by the actual count.

NO Ignore this error condition.

Phase 5: Check Cylinder Groups

This phase concerns itself with the free-block maps. This section lists error conditions resulting from allocated blocks in the free-block maps, free-blocks missing from free-block maps, and the total free-block count not matching the count contained in the summary information area.

CG ... : BAD MAGIC NUMBER

The magic number of the cylinder group is wrong. This usually indicates that the cylinder group maps have been destroyed. When running manually, the cylinder group is marked as needing to be reconstructed. If you are preening your system, this error is fatal.

EXCESSIVE BAD BLKS IN BIT MAPS (CONTINUE)?

You should never get this message. If you do, contact your local HP Sales and Service office.

SUMMARY INFORMATION *t* BAD

where *t* is one or more of:

(INODE FREE)

(BLOCK OFFSETS)

(FRAG SUMMARIES)

(SUPER BLOCK SUMMARIES)

The indicated summary information was found to be incorrect. This error condition will always invoke the “BAD CYLINDER GROUPS” condition in phase 6. If you are preening, the summary information is recomputed.

x BLK(S) MISSING

A number of blocks *x* that are unused by the file system were not found in the free-block maps. This error condition always invokes the “BAD CYLINDER GROUPS” condition in phase 6. If you are preening, the block maps are rebuilt.

FREE BLK COUNT WRONG IN SUPERBLOCK (FIX)?

The actual count of free blocks does not match the count in the superblock of the file system. If you are preening, the counts are fixed.

Possible responses to the “FIX” prompt are:

YES Replace the count in the superblock by the actual count.

NO Ignore this error condition.

BAD CYLINDER GROUPS (FIX)?

Phase 5 has found bad blocks in the free-block maps, duplicate blocks in the free-block maps, or blocks missing from the file system. If you are preening, the cylinder groups are reconstructed.

Possible responses to the “FIX” prompt are:

YES Replace the actual free-block maps with new free-block maps.

NO Ignore this error condition.

Phase 6: Salvage Cylinder Groups

This phase concerns itself with reconstructing the free-block maps. No error messages are produced.

Cleanup

Once a file system has been checked, a few cleanup functions are performed. This section lists advisory messages about the file system and modification status of the file system.

f files, *b* used, *r* free (*y* frags, *z* blocks)

This message indicates that the file system just checked has a total of *f* files, using *b* fragment-sized blocks, with *r* fragment-sized blocks available (free) for use. The numbers in parentheses divides the free count into *y* free fragments and *z* free full sized blocks.

No action is required on your part.

***** REBOOT HP-UX; DO NOT SYNC (USE reboot -n) *****

This message indicates that the root file system has been modified by `fsck`. If HP-UX is not rebooted immediately, the work done by `fsck` can be undone by the in-core (memory) copies of tables HP-UX keeps. If you are preening, `fsck` exits with a code of 4. The `bcheckrc` script interprets an exit code of 4 by executing the `reboot` command.

***** FILE SYSTEM WAS MODIFIED *****

This message indicates that the current file system was modified using `fsck`. If this file system is mounted, `fsck` exits and you should reboot your system. If the system is not rebooted immediately, the changes made by `fsck` might be undone by the memory tables of HP-UX.

Note

If you are preening, you will not get this message. If you execute `fsck -p` outside the `bcheckrc` program, you must check the return code to see if the system needs to be rebooted.



Glossary

alternate boot path

Stored in a special place in the memory of a Series 700 or Series 800 computer, the alternate boot path is the hardware address of a device from which you occasionally (but not usually) boot your system. This is often a tape drive. See also **primary boot path**.

autoboot

A flag located in a special place in the memory of a Series 700 or Series 800 computer. When set, the AUTOBOOT flag tells the computer that it should automatically attempt to boot itself when the computer is powered on (or reset) using the string of characters in the autoexecute file.

autocreation

If you create a CDF, but specify only the path name of the CDF and not a subfile, the system automatically creates a CDF subfile named after the cluster node name attribute. This is known as autocreation. See the manual *Managing Clusters of HP 9000 Computers* for details.

boot area

A special area on bootable disks that contains the LIF volume header, the directory that defines the contents of the volume, and programs that are used by the system during the start-up process.

On Series 300 and Series 400 computers, eight kilobytes of disk space are reserved at the beginning of bootable disks. The secondary loader program is located here.

On Series 700 computers and Series 800 computers, ISL, the autoexecute file (AUTO), the "hpux" loader utility and other files related to booting the system are located here.

Glossary

On Series 800 computers using the Logical Volume Manager for their root file systems, and on Series 800 computers using the SwitchOver/UX product, the LABEL file is stored in the boot area.

boot rom

The **boot ROM** is a small, machine language program that resides in your computer's read-only memory. When you boot or re-boot the system, the computer starts the boot ROM program, which takes control of the system. The boot ROM tests computer hardware, finds some devices accessible through the computer, and loads an operating system. (See Chapter 2, "System Startup" in the manual *How HP-UX Works: Concepts for the System Administrator*, or Chapter 5, "System Boot-Up Problems" in this manual.)

booting

The process of starting up the HP-UX operating system.

CDFS

The CDFS (for CD ROM File System) is used on compact disks to implement the HP-UX directory structure.

clients

See **cluster client**.

cluster client

A cluster node that does not have the root file system for the cluster on its local disks. Its root file system resides on the cluster server. Cluster client computers can have locally-mounted file systems (other than the root file system).

cluster node

A computer in a cluster. See the manual *Managing Clusters of HP 9000 Computers* for details.

cluster related parameters

Kernel parameters associated only with HP-UX clusters.

cluster server

The cluster node that acts as the root-file-system server for all the clients

in an HP-UX cluster. See the manual *Managing Clusters of HP 9000 Computers* for details

cnodes

See **cluster node**.

context

In HP-UX clusters, context refers to a specific element in a context dependent file. A context string is used to determine which element of a context-dependent file will be selected. See **context string**.

context-dependent files

Also known as “hidden directories,” context-dependent files are files having different contents, depending on which cluster node uses them. These are actually directories disguised as files. The files within these special directories contain the various contents (contexts) of the context-dependent file. See also, **context**, and **context string**.

context string

Used with HP-UX clusters. Every HP-9000 computer has a string that identifies the following context types:

- Its cluster node name (the string “standalone” is used for computers that are not part of a cluster).
- Its floating-point hardware type (for example, HP98248A, HP-MC68881, HP98635A, etc.)
- Its processor type (for example, HP-MC68020, HP-PA, PA-RISC1.1)
- Its cluster node type (`localroot` or `remoteroot`)
- The string “default”.

EXAMPLE:

```
hpulpcu4 PA-RISC1.1 HP-PA localroot default
```

When access to a context-dependent file is needed, the elements of the context string are used (from left to right) to search the elements of the context-dependent file until a match is found. Once a match has been found, the contents of that file (CDF element) become the “context” for the

current CDF access. For details, see Chapter 2, “Understanding Clusters”, in the manual *Managing Clusters of HP 9000 Computers*.

context types

See **context string**.

data block

A disk block containing the data for a file. See also, **disk block**.

direct connection

In the UUCP subsystem, a direct connection is one that is made without using a modem. This is typically accomplished using a serial cable between two computers.

disk block

A fixed size unit of disk space; part of a file system. There are several types of disk blocks: data blocks, inodes, indirect blocks, and superblocks. Disk blocks are composed of **fragments**.

disk partition

See **disk section**.

disk section

A predefined, fixed-size area of a disk, appearing to the operating system as if it were a separate disk. See also **logical volume**.

extent

Fixed-size addressable areas of space on an LVM disk or in memory. On disk, these areas are called physical extents. Physical extents correspond to areas in memory called logical extents. See Chapter 9, “Logical Volume Manager” in the manual *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029 for details. If disk mirroring is *not* used, there is a one-to-one relationship between the two types of extents. If disk mirroring is used, each logical extent corresponds (maps) to more than one physical extent.

file system

An organization of files and directories on a particular device, disk section, or logical volume: used to build the HP-UX directory structure.

The HFS (for High-Performance File system) is typically used on hard disks, disk sections, and logical volumes to implement the directory structure.

The CDFS (for CD ROM File System) is used on compact disks to implement the directory structure.

The NFS (for Network File system) is used to access file systems on other computers, over a network.

fragment

A piece of a disk block. This is the smallest unit of information that the High Performance File System will read or write. The lower limit of a fragment's size is `DEV_BSIZE` (defined in `/usr/include/sys/param.h`). Fragment size is set at file system creation and is constant throughout a given file system. See Chapter 8, "HFS File System" in the manual *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029 for details.

FSCK

A utility used to check the integrity of an HP-UX file system.

getty

A process responsible for coordinating the login process. Getty's are typically started by the "init" process. They display the login prompt and wait for a user to attempt to log in. Getty's start up the login program when they detect activity on the terminal.

hidden directories

See **context-dependent files**.

HP-UX

The operating system used by HP9000 computers.

hpux

An HP-UX specific loader program scheduled by ISL (the Initial System Loader) to load and start up HP-UX.

/hp-ux

The file in the root file system usually used to contain the HP-UX kernel.

indirect block

A disk block that contains pointers to the data blocks of a file, or to other indirect blocks. In large files, pointers to the first few data blocks of the file are stored in the file's **inode**. If the file requires more data blocks than its inode can point to, the inode can point to an indirect block, which has room for additional pointers.

initial system loader

The Initial System Loader (ISL) is the first piece of software that runs during the boot-up process. It is loaded and started by the Boot ROM program. ISL runs the hpux utility that is used to load the HP-UX operating system.

inode

A data structure containing information about a file such as file type, pointers to data, owner, group, and protection information. See Chapter 8, "HFS File System" in the manual *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029 for details.

ISL

See **Initial System Loader**.

line printer spooling system

An HP-UX subsystem: used for routing printer output, preventing mixed listings, and controlling the flow of data to the printers on an HP-UX system.

logical extent

A set of virtual blocks of data contained within a logical volume. A logical extent corresponds (maps) to one (more, if LVM mirroring is used) physical extent.

logical volume

A *logical* (rather than physical) construction that is a map of data stored on LVM disks (physical volumes). A logical volume can be conceptualized as a storage device of flexible size. The data in a logical volume can be located on one or more physical volumes (disks).

lost+found directory

A special directory used by the “fsck” utility to store files and directories for which fsck could not find the proper location.

lpsched

The utility used to start up the line printer spooling system (also known as the line printer scheduler).

lpshut

The utility used to stop the line printer spooling system (also known as the line printer scheduler).

major number

An index into a device driver table in the kernel. It is needed to communicate with peripheral devices. See Chapter 11, “System Configuration”, in the manual *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029, for details.

map file

A file written by the vgexport command and read by the vgimport command. The map file is used to transfer logical volume names and numbers between LVM systems.

minor number

Part of a device file; a hexadecimal number that contains driver-specific information. The minor number often contains device addressing information such as a SCSI address, multiplexer port number, etc.

mirror allocation policy

Allocation refers to how mirrored copies of data are distributed to LVM disks (physical volumes). The allocation policy can be:

- | | |
|------------|--|
| strict | Copies of the original data cannot reside on the same physical disk as the original. |
| non-strict | Copies of the original data can reside on the same physical disk as the original data. |
| contiguous | No gaps are permitted in the set of physical extents on a physical volume. |

non-contiguous Gaps are permitted in the set of physical extents on a physical volume.

See “Managing Logical Volumes” in the *System Administration Tasks* manual, and Chapter 9, “Logical Volume Manager” in the manual *How HP-UX Works: Concepts for the System Administrator* for more details.

mirror consistency recovery (MCR)

A method of ensuring data consistency following a system crash or power failure. Although recovery times will be longer than when using the Mirror Write Cache mechanism, performance during normal system operation will not be degraded, as with the Mirror Write Cache mechanism. See also, **Mirror Write Cache (MWC)**

mirror write cache (MWC)

A MirrorDisk/UX mechanism, whose use is optional, that tracks outstanding mirror write requests and provides a basis for resynchronization of data blocks after a system crash or power failure. The use of the MWC degrades performance, as extra work is required during disk writes to maintain the Mirror Write Cache.

mirroring

Replication of data using an optional product, MirrorDisk/UX. This capability ensures a greater degree of data availability. Mirroring maps logical extents to multiple physical extents, thus providing the means to recover easily from the loss of one copy (or two copies in the case of three-way mirroring) of data. Mirroring can provide faster access to data for database applications using more data reads than writes. See Chapter 9, “Logical Volume Manager” in the *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029 for more information about mirroring.

MWC

See **Mirror Write Cache (MWC)**.

panic

An unrecoverable system failure, often caused by a hardware failure, or by a lack of system resources. HP-UX reached a point where it could not proceed.

panic message

A message printed on the system console indicating that HP-UX has panicked. See Chapter 10, "System Panics" in this manual for details. See the *Error Messages Catalog* for information on specific panic messages.

physical extent

A specific, contiguous set of blocks within a region of a physical volume (LVM disk), where data resides. Physical extents are a consistent size within a specific volume group. They can range in size from one megabyte to 256 megabytes (but their size, in megabytes, must be a power of two).

physical volume

A disk containing LVM data structures (an LVM disk). A physical volume is an entire disk drive; therefore, the device files associated with section 2 for the disk are used.

primary boot path

Stored in a special place in the memory of a Series 700 or Series 800 computer, the primary boot path is the hardware address of the device from which you normally boot your system.

priority fence

Associated with printers in the line printer spooling system, a priority fence defines the minimum priority that a print request must have before it will be allowed to print on a given printer.

quorum

The number of physical volumes required to be available to activate a volume group, and/or maintain the Mirror Write Cache (MWC).

request directories

Directories used by the line printer spooling system to store print requests while they are printing (or waiting to be printed).

request-ID

A unique identification number used to identify a specific print request in the line printer spooling system.

root file system

The file system that contains the top of the HP-UX directory tree. It is the first file system mounted during the boot sequence.

secondary loader program

On a Series 300 or Series 400 computer, this is the piece of software that is loaded by the boot ROM program. The secondary loader program locates, loads, and runs the HP-UX operating system.

superblock

A data structure containing global information about a file system such as the file system's size, disk information, and cylinder group parameters. The superblock is created at the same time as the file system and is replicated into each cylinder group. Also, HP-UX keeps a copy of the superblock in memory at all times. The `sync` command writes the superblock to the disk. See Chapter 8, "HFS File System" in the manual *How HP-UX Works: Concepts for the System Administrator*, HP part number B2355-90029 and `sync(1M)` in the *HP-UX Reference Manual* for details.

system default destination

When no printer is specified with a print request, the printer defined to be the system default destination (if one has been defined) will be used by the line printer spooling system to print the request.

system panic

See **panic**.

uucico

The "Unix to Unix Copy In Copy Out" utility is the underlying program in the UUCP subsystem used to actually transfer data from one computer to another.

UUCP

A protocol, common to most UNIX systems, used for transferring data from one computer to another, usually over serial lines.

volume group

The combined disk space of one or more physical volumes (LVM disks), from which logical volumes are allocated.

Index

A

- activating volume groups, 8-13, 8-19, 8-20, 8-21
- adding disks, 7-9
- alternate boot path, 5-19, Glossary-1
- archiving files, 7-7
- attended mode (boot), 5-23
- autoboot, 5-18, 5-24, Glossary-1
- autoboot flag, 5-17
- autocreation, 4-7, Glossary-1
- autosearch flag, 5-17

B

- backing up
 - LVM configuration information, 8-7
- backups, 7-7, 10-6
 - importance of, 1-9, 10-1
 - utilities, 7-7
- block, duplicate, A-11
- boot area, 8-6, 8-15, Glossary-1
- boot devices
 - list of potential, 5-15
- booting, Glossary-2
- boot process
 - phases of, 5-1
 - summary, 5-1
- boot ROM, 5-8, 5-11, Glossary-2
- boot ROM search order (Series 300/400 computers), 5-12
- boot-up
 - attended mode, 5-23
 - autoboot, 5-18

- following system panic, 10-5
 - from LAN, 5-30
 - from LVM disks, 8-12
 - kernel selection, 5-20, 5-25
 - LVM-related problems, 8-11
 - maintenance mode, 8-16
 - missing physical volumes, 8-12
 - override autoboot, 5-24
 - overriding quorum requirement, 8-14
 - single-user mode, 8-16
 - unattended mode, 5-23
 - wrong kernel, 5-12, 5-23
- boot-up problems, 5-1
 - quick reference table, 5-2

C

- cannot activate volume groups, 8-19, 8-21
- cannot boot LVM system, 8-11-19
- can't log in, 5-30
- CDF autocreation, 4-7
- CDF indicator, 4-3
- CDF Mechanism, 4-3
- CDF Mix-ups, 4-7
- CDFS, Glossary-2
- CD ROM File System, Glossary-2
- check blocks and sizes, A-7
- check connectivity, A-17
- check cylinder groups, A-21
- check reference counts, A-18
- client, 4-1, Glossary-2
- cluster

- autocreation, 4-7
- CD-ROM compatibility, 4-6
- client, 4-1
- cnodes, 4-1
- context, 4-10
- file system problems, 6-4
- getcontext command, 4-3
- hidden directories, 4-1, 4-3
- kernel parameters, 4-5
- Link Level Address, 4-10
- log files, 4-12
- makecdf, 4-10
- NFS compatibility, 4-6
- rbootd, 4-4
- remote boot, 4-4
- server, 4-1
- showcdf, 4-10
- swap space, 4-1, 4-6
- Cluster
 - context, 4-1
- cluster server, Glossary-2
- cnodes, 4-1
- command
 - accept, 2-1
 - adb, 7-6
 - bdf, 7-5, 8-39
 - cancel, 2-3, 2-6, 2-7
 - cat, 2-10, 2-11, 3-9, 6-15, 9-7
 - ccck, 4-5, 4-11
 - cdb, 7-6
 - cp, 7-8
 - cpio, 7-7
 - cu, 3-1, 3-12, 3-14, 3-15
 - dd, 7-7, 8-5, 8-8
 - disable, 2-1, 2-5
 - diskinfo, 8-29
 - dump, 7-7
 - enable, 2-1, 2-5
 - extendfs, 8-22, 8-23, 8-26
 - fbackup, 7-7
 - find, 4-10
 - frecover, 7-7
 - fsck, 6-8, 10-4, 10-6, A-1
 - fuser, 6-9, 6-17
 - getcontext, 4-3
 - getty, 9-7
 - grep, 8-34, 8-37, 8-38
 - hpux, 8-14, 8-18
 - init, 4-10, 4-11, 5-29, 9-2, 9-7
 - ioscan, 8-30
 - kermit, 3-11
 - kill, 6-9, 9-6
 - ll, 2-12, 3-3, 4-10, 9-8
 - ln, 7-8
 - lp, 2-6, 2-10
 - lpadmin, 2-3, 2-12
 - lpmove, 2-3, 2-7, 2-8
 - lpsched, 2-3, 2-7, 2-8, 2-10
 - lpshut, 2-7, 2-8, 2-10
 - lpstat, 2-6, 2-11
 - ls, 4-3, 4-10
 - lssf, 2-12, 3-3, 9-8
 - lvdisplay, 8-37, 8-38, 8-42
 - lvextend, 8-22, 8-23
 - lvlnboot, 8-13, 8-14, 8-15, 8-18
 - makecdf, 4-10
 - mkboot, 8-6, 8-17
 - mkfs, 7-6
 - more, 6-15
 - mount, 6-8, 6-16
 - newfs, 7-6, 8-8
 - ps, 3-14, 9-3
 - pvcreeate, 8-2
 - rbootd, 4-4, 5-15
 - reboot, 10-4, 10-6
 - reboot, -n option, 10-4, 10-6
 - reject, 2-1, 2-7
 - restore, 7-7
 - rlp, 2-3
 - rm, 7-6
 - script, 1-4
 - set (*shell command*), 9-11

- showcdf, 4-10
 - strings, 6-15
 - stty, 3-11, 9-4, 9-10
 - swapinfo, 8-38
 - tar, 7-7
 - tcio, 7-7
 - telinit, 9-8
 - truncate, 6-5
 - umount, 6-10, 6-17
 - uname, 8-28
 - uuccheck, 3-21
 - uucico, 3-2, 3-17, 3-19
 - uucp, 3-1, 3-17
 - uulog, 3-19
 - uuls, 3-21
 - uustat, 3-21
 - uux, 3-1
 - vgcfgbackup, 8-7, 8-8
 - vgcfgrestore, 8-7, 8-9
 - vgchange, 8-14, 8-20, 8-21
 - vgdisplay, 8-34, 8-37
 - vgexport, 8-52, 8-53
 - vgimport, 8-52, 8-53
 - vgscan, 8-20, 8-50
 - what, 6-15
 - who, 9-2
 - common mistakes
 - when using the logical volume manager, 8-8
 - computer monitor, 5-8
 - connecting cables, 9-9, 10-3
 - connectivity, check, A-17
 - consistency check, A-1
 - console screen, 5-8
 - context, 4-1, 4-10, Glossary-3
 - context string, 4-3
 - types of, 4-2
 - context-dependent files, Glossary-3
 - Context-Dependent Files, 2-4, 4-1, 5-29
 - /dev, 4-2
 - /etc/btmp, 7-8
 - /etc/inittab, 4-2
 - /etc/utmp, 7-8
 - /etc/wtmp, 7-8
 - /hp-ux, 4-2
 - /usr/bin/cancel, 2-4
 - /usr/bin/slp, 2-4
 - /usr/lib/disable, 2-4
 - /usr/lib/lpsched, 2-4
 - context string, Glossary-3
 - context types, Glossary-4
 - copying data
 - using the dd command, 8-8
 - core dump, 7-6
 - core files, 10-5
 - cylinder groups, A-21
 - cylinder groups, salvaging, A-22
- ## D
- daemon
 - remote boot (rbootd), 5-15
 - data blocks, Glossary-4
 - dd command
 - caution with LVM disks, 8-5
 - device files, 2-11, 3-3, 3-5, 3-6, 3-7, 3-9, 3-11, 6-9, 9-4
 - Device Files, 9-7
 - dialit.c, 3-14
 - direct connection, Glossary-4
 - disk block, Glossary-4
 - disk blocks
 - buffer caching, 6-3
 - fragments, 6-1
 - disk mirroring, 8-1
 - disk organization
 - Series 300 computers, 7-1
 - Series 800 computers, 7-1
 - disk partition, Glossary-4
 - disk partitions, 8-2
 - disk section, Glossary-4
 - disk sections, 7-1, 7-2, 8-2
 - vs. logical volumes, 8-2

Index

disk space
 organization, 7-1
 shortages, 7-1
 superuser's extra,, 7-6
 system performance, 7-6
Disk Space
 shortage, 7-5
duplicate block, A-11

E

error messages
 Device Busy, 6-10
 during boot-up, 4-11
 Failed kernel selftest, 4-11
 fsck, A-7
errors
 initialization phase, A-4
extent, Glossary-4
extents, 8-3

F

file
 core, 7-6, 10-5
 /etc/bcheckrc, 5-29
 /etc/brc, 5-29
 /etc/btmp, 7-8
 /etc/clusterconf, 4-5, 4-10, 4-11,
 5-31
 /etc/disktab, 7-3
 /etc/gettydefs, 9-10
 /etc/inittab, 3-5, 3-6, 5-29, 9-7,
 9-8
 /etc/lvmconf, 8-8
 /etc/lvmtab, 8-5, 8-8, 8-13, 8-20,
 8-21, 8-50, 8-51, 8-52
 /etc/mnttab, 6-16
 /etc/passwd, 3-5, 3-6, 3-7
 /etc/rc, 5-29, 6-16
 /etc/utmp, 7-8
 /etc/vgsync, 8-40
 /etc/wtmp, 7-8

/hp-ux, 5-11, 8-14, 8-18
LIF:LABEL, 8-17, 8-18
S800, 8-19
/tmp/cluster.log, 4-12
/usr/adm/errorlog, 4-12
/usr/adm/rbootd.log, 4-12
/usr/lib/uucp/Devices, 3-7, 3-8,
 3-12
/usr/lib/uucp/Dialers, 3-14
/usr/lib/uucp/Permissions, 3-18,
 3-21
/usr/lib/uucp/Systems, 3-5, 3-6,
 3-7
/usr/lib/uxbootlf, 8-6
/usr/spool/uucp/DIALLOG, 3-21

files

 archiving, 7-7
 corrupt, 6-12
 corrupted, 6-5, 6-6
 determining file system, 6-8
 missing, 6-5, 6-12
 moving, 7-7
 ownership, 6-2, 6-12
 permissions, 6-2, 6-12
 removing, 7-6
file system, 6-1, 7-1, Glossary-4
 backups, 6-13
 corruption, 6-3, 6-5, 6-13
 corruption (symptoms of), 6-4
 data blocks, 6-3
 disk blocks, 6-1
 fragments, 6-1
 inconsistency, 6-3
 indirect blocks, 6-3
 inodes, 6-2
 logical vs. physical layout, 6-1
 lost+found directory, 6-15
 mounting, 6-16
 on logical volume, 8-22
 orphaned files, 6-15
 root, 7-1

- structure of, 6-1
- superblock, 6-1
- syncer, 6-3
- system panic, 10-3
- unmounting, 6-10

file system consistency check (fsck), A-1

flags

- autoboot, 5-17
- autosearch, 5-17

fragment, Glossary-5

fsck, 5-29, **6-5**, Glossary-5

- errors, 6-13
- interactive mode, 6-13
- interpreting results, 6-11
- n option, 6-14
- preening mode, 6-11
- quiescent file system, 6-8
- use with caution, 6-14
- y option, 6-14

G

garbage data

- on printers, 6-6
- on terminals, 9-1, 9-9

getcontext, 4-3

getty, 9-7, 9-10, Glossary-5

gettydefs, 9-7, 9-10

H

hardware

- device address, 2-11
- failure at boot-up, 5-10, 10-2
- failure of peripheral, 10-3
- problem, 3-2
- terminals, 9-8
- UUCP connections, 3-8

hardware address, 2-11

hidden directories, **4-1**, 4-3, 4-10, Glossary-5

- contents of, 4-3

HP LVM Mirroring, Glossary-8

HP-UX, 5-11, Glossary-5

hpux, 5-11, Glossary-5

- “-lq” option, 8-14

HP-UX runstate, 9-2

HP-UX version number, 1-2

hung programs, 8-26

I

identifying runstate, 9-2

improper system shutdown, 6-3

inconsistencies, file system, A-4

indirect block, Glossary-6

init, 4-11, 5-29

initialization phase errors, A-4

initial system loader, Glossary-6

inittab entries, 9-8

inode, Glossary-6

inode, root, A-12

inodes, 6-2

intermixed listings, 2-1

ISL, Glossary-6

ISL (Initial System Loader), 5-11

K

kernel parameters, 4-5

killing processes, 9-6

L

LABEL file, 8-17, 8-18

LAN, 4-1, 4-7

- booting from, 5-15
- hardware connection, 5-15
- Link Level Address, 4-10
- problems with, 10-4
- system panics, 10-4

LED (front panel) display, 5-10

LIF area

- LABEL file, 8-17, 8-18
- location of, 8-6
- on LVM disks, 8-6, 8-17

LIF (Logical Interchange Format), 5-11

- LIF volume
 - on Series 300/400 computers, 5-11
 - on Series 700 computers, 5-11
 - on Series 800 computers, 5-11
- lights
 - front panel, 5-8, 5-10
- line printer spooling system, 2-1
 - commands, 2-2
 - components of, 2-1
 - error messages, 2-12
 - paper jams, 2-4
 - paper out, 2-4
 - printer names, 2-11
 - priority fence, 2-10
 - remote spooling, 2-3
 - scheduler, 2-4, 2-7
 - scheduler status, 2-12
 - status of, 2-6, 2-8, 2-11
- linking, 7-7
- Link Level Address, 4-10
- local swap space, 4-1
- lock files, 3-14
- log book, 1-2
 - file system example, 6-6
 - sample form, 1-3
- log files, 1-4, 7-8
 - DIALOG, 3-21
 - HP-UX Clusters, 4-12
 - UUCP, 3-19
- logical extent, Glossary-6
- logical volume, Glossary-6
- logical volume implementation, 8-3
- Logical Volume Manager, 7-3, 8-1
- logical volumes, 8-1-49
 - file systems on, 8-22
 - increasing size of, 8-21
 - reducing size of, 8-5, 8-26
 - swap space on, 8-23, 8-24
 - vs. disk sections, 8-2
- log in
 - can't, 5-30
- lost+found directory, 6-15, A-3, A-19,
Glossary-7
- lpsched, Glossary-7
- lpshut, Glossary-7
- LVM
 - pointers to root, swap, dumps areas,
8-11, 8-14, 8-18
 - user data space, 8-5
- LVM configuration information
 - backing up, 8-7
 - restoring, 8-8, 8-9
- LVM data structures, 8-16
 - corrupted, 8-15
- LVM problem prevention, 8-4
- LVM problems
 - cannot boot system, 8-11
 - not enough disks present to boot,
8-11, **8-12**
 - preparing for, 8-7
- LVM system configuration
 - picture of, 8-9
- lvmtab file, 8-5, 8-8, 8-20
 - missing, 8-21
 - recovering, 8-50
- M**
- maintenance mode boot, 8-16
- major number, 9-7, Glossary-7
- map file, Glossary-7
- MCR, Glossary-8
- minor number, 2-12, 9-7, Glossary-7
- mirror allocation policy, Glossary-7
- Mirror Consistency Recovery
(mechanism), Glossary-8
- MirrorDisk/UX, 8-1
- mirroring, 8-1, Glossary-8
- Mirror Write Cache, Glossary-8
- mkboot command, 8-6
- modem, 3-3
 - commands, 3-9
 - configuration, 3-10

- settings, 3-10
 - transmit/receive indicators, 3-9
 - UUCP connections, 3-8
 - monitor, 5-8
 - moving files, 7-7
 - multi-user mode, 9-2
 - MWC, Glossary-8
- N**
- no power to computer, 5-8
- O**
- orphaned files, 6-15
 - overriding quorum requirement, 8-13, 8-15, 8-21
 - ownership, 6-2
- P**
- panic, Glossary-8
 - panic message, 10-2, Glossary-9
 - path name, A-11
 - PDC (Processor Dependent Code), 5-8
 - permissions, 3-14, 3-19, 6-2
 - physical extent, Glossary-9
 - physical volume, 8-2, Glossary-9
 - physical volumes
 - removing from a system, 8-7
 - removing from a volume group, 8-7
 - picture of LVM system
 - how to create, 8-27-49
 - information to include, 8-10
 - reason for, 8-9
 - preening mode, A-1
 - preparing for LVM problems, 8-7
 - primary boot path, 5-19, Glossary-9
 - printers, 2-3
 - device files, 2-11
 - hardware address of, 2-11
 - problems with, 2-10
 - won't print, 2-9
 - print jobs
 - restarting from the beginning, 2-5
 - restarting from where it stopped, 2-6
 - print requests
 - canceling, 2-6
 - moving, 2-7
 - priority of, 2-10
 - request-id, 2-6, 2-8
 - routing of, 2-3
 - won't print, 2-4
 - priority fence, 2-10, Glossary-9
 - problem definition, 1-1, 1-4
 - file system problems, 6-4
 - printer won't print, 2-9
 - things to consider, 1-5
 - what's different, 1-5
 - problem isolation, 1-1, 1-6
 - printer won't print, 2-9
 - UUCP problems, 3-8
 - problems
 - booting from LAN, 5-30
 - booting wrong kernel, 5-23
 - CDF Mix-ups, 4-7
 - cluster client won't boot, 4-4
 - cluster problems, 4-4
 - cluster server won't boot, 4-4
 - disk space shortages, 7-6
 - file permissions, 3-2
 - identifying possible causes, 1-7
 - mounting file systems, 6-16
 - preparing for, 1-8
 - preventive measures, 1-8
 - system boot-up, 5-1, 10-3
 - testing possible causes, 1-7
 - unmounting file systems, 6-10, 6-17
 - UUCP configuration, 3-2
 - with hardware, 3-2, 10-2
 - with LAN, 4-7
 - with printers, 6-7
 - with terminals, 9-1, 9-9
 - processes
 - associated w/terminals, 9-3

- terminating, 6-8, 6-9, 9-6
 - with files open, 6-8
- processor
 - resetting, 5-10
- program hangs, 8-26
- protecting LVM configuration
 - information, 8-7
- pvcreate, 8-2

Q

- quorum, 8-12, 8-19, 8-20, Glossary-9

R

- rbootd, 4-4
- record keeping, 1-2
 - benefits, 1-2
 - HP-UX version number, 1-2
 - online vs. notebook, 1-4
 - tools, 1-4
 - what information, 1-2
- reducing size of logical volumes, 8-5
- reference counts, A-18
- remote boot, 4-4
- remote boot daemon (rbootd), 5-15
- removing files, 7-6
- request directories, Glossary-9
- request-ID, Glossary-9
- rescan for dups, A-11
- reset button, 5-10
- restoring
 - LVM configuration information, 8-8
- rlpdaemon, 2-3
- root file system, Glossary-10
 - corrupted, 8-16
 - on a logical volume, 8-14
 - on LVM system, 8-11
 - out of date, 8-14
 - restoring, 8-14
- root inode, A-12
- runstate, 9-2

S

- salvage cylinder groups, A-22
- SAM, 3-2, 4-7
 - modify kernel parameters, 4-5
- SEARCHING FOR A SYSTEM (Return to pause), 5-12
- secondary loader program, 5-11, 5-12, 5-20, Glossary-10
- sections, disk, 7-2
- server, 4-1
- SETUID bit, 4-10
- single-user mode, 8-16, 9-2
- superblock, 6-1, Glossary-10
- swap space, 4-1, 4-6
 - on logical volumes, 8-23, 8-24
- symbolic links, 7-7
- system configuration
 - picture of, 8-9
- system default destination, 2-11, Glossary-10
- system panic, Glossary-10
 - hardware failure, 10-2
- system panics, 10-1
 - after reducing size of logical volume, 8-23, 8-25
 - HP-UX cluster specific, 4-6
 - LAN, 10-4
 - recovering from, 8-23, 8-25, 10-1
- system runstate, 9-2

T

- terminal configuration, 9-5
- terminal, resetting, 9-4
- terminals, 9-1, 9-9
- terminating processes, 9-6
- TOC button, 5-10
- tools, 3-19
 - working with CDF's, 4-8
- traditional disk sections, 8-1-3
- transfer-of-control button, 5-10
- troubleshooting techniques, 1-1

U

unattended mode (boot), 5-23
 unresponsive terminals, 9-1
 user data space
 LVM, 8-5
 uucico, Glossary-10
 UUCP, Glossary-10
 background processes, 3-2
 behind the scenes, 3-17
 cable wiring, 3-12
 communication layers, 3-8
 configuration files, 3-2
 device files, 3-5
 Dialers file, 3-14
 direct connection, 3-3
 direct connections, 3-11, 3-14
 file transfer, 3-1, 3-8, **3-17**
 information flow, 3-16
 log files, 3-19
 modem connection, 3-3
 modem connections, 3-8, 3-12
 Permissions file, 3-18
 remote command execution, 3-1, 3-8,
 3-17
 remote login, 3-1, 3-8
 status information, 3-19
 testing "Devices" file, 3-12
 testing "Systems" file, 3-15

uucheck, 3-21
 uucico, 3-17, 3-19
 uucp, 3-18
 uucp file transfer, 3-17
 uulog, 3-19
 uuls, 3-21
 uustat, 3-21
 Uutry script, 3-19
 versions of, 3-1
 uxbootlf file, 8-6

V

vgcfgbackup command, 8-7
 vgcfgrestore command, 8-7
 vgexport, 8-53
 vgimport, 8-53
 vgscan, 8-51
 -p to preview, 8-51
 running to update `lvmtab`, 8-51
 volume group, **8-3**, Glossary-10
 volume groups
 activating, 8-13, 8-19, 8-20, 8-21
 cannot activate, 8-19, 8-21
 moving disk between, `vgimport`,
 `vgimport`, 8-52
 overriding quorum requirement, 8-13,
 8-15

Reorder No. or
Manual Part No.
B2355-90030

Copyright © 1992
Hewlett-Packard Company
Printed in USA E0892

**Manufacturing
Part No.
B2355-90030**



B2355-90030