

**Sys5 Administrator's Handbook**

98-05133.2 Ver. B

October, 1987

**Sys5 Administrator's Handbook**

98-05133.2 Ver. B

October, 1987

**PLEXUS COMPUTERS, INC.**

3833 North First Street

San Jose, CA 95134

408/943-9433

Copyright 1987  
Plexus Computers, Inc., San Jose, CA

All rights reserved.

No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, without the prior written consent of Plexus Computers, Inc.

The information contained herein is subject to change without notice. Therefore, Plexus Computers, Inc. assumes no responsibility for the accuracy of the information presented in this document beyond its current release date.

Printed in the United States of America

## PREFACE

---

This manual, *Sys5 Administrator's Handbook*, is written for a system administrator with a UNIX<sup>†</sup> background. This manual explains how to use Sys5 to set up and configure your Plexus<sup>®</sup> P/35, P/55, P/60, P/75, P/90, or P/95 system for users. It is not written for casual Sys5 users.

Before you begin, please read the document *Read Me First* which provides:

- an overview of manuals shipped with your Plexus system
- abstracts of each manual
- information on additional publications which can be ordered from Plexus

Before reading this manual, you should have your system installed. See your system's Installation Guide for installation instructions. This manual will refer you to additional manuals comprising the A T & T manual set. Plexus also provides Maintenance Manuals for your Plexus system. The Maintenance Manuals are designed and written for service personnel and original equipment manufacturers (OEMs) and provide specific information regarding the components and maintenance of your Plexus system. Following is a list of manuals that might be of interest to you:

---

† UNIX is a registered trademark of AT&T.

® Plexus is a registered trademark of Plexus Computers, Inc.

- 
- your latest Sys5 Release Notice
  - the Maintenance Manual for your system
  - Volumes comprising the UNIX Sys5 set. These volumes are documented in *Read Me First*.
  - *Sys5 UNIX Error Message Reference Manual*

## Notation Conventions

The following notation conventions are used in this manual:

**bold face** is used for commands you enter exactly as shown (i.e. **ps -aef**), for command and utility names (i.e. **cpio(1)**), and to show what you would type at your terminal keyboard.

*italic* is used for commands for which you must substitute a value. For example, in the following command **kill -9 XXXX** you would substitute XXXX with the actual process number to be killed (such as **3271**). Italic type is also used for manual titles, (i.e. *Sys5 UNIX User's Reference Manual*).

typewriter is used for displays the system returns such as:  
Initializing VCP0

Instructions to enter commands are set off from the rest of the text and are in **bold face** type. Press the RETURN key after each command line unless instructed otherwise. Numbers in parentheses after a command name — such as **dnld(1)** indicate the section of the *Sys5 UNIX Reference Manual* in which documentation for that command can be found.

---

## References

Although this manual is meant to be comprehensive in scope, additional information is always helpful. Following is a list of commercially available books and periodicals on UNIX that contain useful information for system administrators.

Fiedler, David and Hunter, Bruce H. *UNIX System Administration*. Hayden Book Company: Hasbrook Heights, New Jersey, 1986.

This book is an excellent presentation of how to perform system administration procedures on a UNIX system. It describes how to perform tasks through step-by-step procedures. This book covers differences between different versions of UNIX (i.e. Sys3 and Sys5).

Foxley, Eric. *UNIX For Super-Users*. Addison Wesley Publishing Company, 1985.

This book is an introductory text on system administration. It contains similar information to this manual. Some of the topics covered in Eric Foxley's book include: the shell, performance considerations, security on a UNIX system, major commands for managers (**make**, **SCCS**, **awk**, and process scheduling programs), and a summary of Bourne shell and C shell features.

Groff, James R. and Weinberg, Paul N. *Understanding UNIX: A Conceptual Guide*. Que Corporation: Indianapolis, Indiana, 1983.

Although this book is not written for system administrators, it contains much useful information. It explains how the file system is organized, describes how to use the shell, contains a small section on system administration, describes process management (**fork** and **exec**), and contains additional chapters on text processing and software development.

---

Kochan, Steven G. and Wood, Patrick H. *Exploring the UNIX System*. Hayden Book Company: Hasbrouck Heights, New Jersey, 1984.

This book is written for UNIX users (not system administrators). However, it does contain a very good chapter on system administration. You should consider buying this book if your knowledge of UNIX is minimal and you need to come up to speed on how to use UNIX quickly.

*Unix World*. Tech Valley Publishing, Mountain View, CA.

This journal is published monthly. It covers different topics each month. Each issue also describes how to perform system administration procedures. A few examples of topics which have been covered include: how to perform backups, how to make your system secure, and how to operate and maintain data communications.

## Revision Record

98-05133.1 Ver A.	Initial release of the document.
98-05133.1 Ver B.	Updated manual to include information specific to the P/95.
98-05133.2 Ver A.	Updated manual to include information specific to the P/90.
98-05133.2 Ver B.	Updated manual to correct errors and include new information specific to the new 1.7 version of UNIX.

## TABLE OF CONTENTS

INTRODUCTION .....	1-1
Before You Begin .....	1-1
Introduction to Plexus Systems .....	1-2
How to Use This Manual .....	1-3
Checklist to Complete .....	1-3
Duties of the System Administrator .....	1-4
Alternate Systems Administrator .....	1-6
System Logbook .....	1-7
Types of Users .....	1-8
Privileged Users .....	1-8
How to Become the Root Superuser .....	1-9
Powers of the Root Super-User .....	1-10
Functional Users .....	1-12
System Passwords .....	1-12
HARDWARE COMPONENTS .....	2-1
Overview .....	2-1
Overview of Plexus Systems .....	2-2
P/35 System .....	2-2
P/35 System Description .....	2-2
P/35 Slot Assignments .....	2-3
Controls and Indicators .....	2-3
Dimensions and Weight .....	2-3
Storage Expansion Module .....	2-4
P/55 System .....	2-5
P/55 System Description .....	2-5
P/55 Slot Assignments .....	2-5
P/55 Storage Expansion Module .....	2-6
Additional Information .....	2-6
P/60 System .....	2-7
P/60 System Description .....	2-7
P/60 Slot Assignments .....	2-7
P/60 Expansion Cabinet .....	2-9
Additional Information .....	2-9
P/75 System .....	2-10
P/75 System Description .....	2-10
P/75 Slot Assignments .....	2-10
P/75 Expansion Cabinets .....	2-11
P/75 Common Circuits Board .....	2-12
Additional Information .....	2-13
P/90 System .....	2-14
P/90 System Description .....	2-14
P/90 Slot Assignments .....	2-15
P/90 Expansion Cabinets .....	2-16
Additional Information .....	2-16
P/95 System .....	2-17
P/95 System Description .....	2-17
P/95 Card Cage Slot Assignments .....	2-17



P/95 Expansion Cabinets.....	2-20
P/95 Arbiter Board.....	2-21
Additional Information.....	2-21
Hardware Comprising Plexus Systems .....	2-22
Backplane .....	2-23
P/90 and P/95 VMEbus Backplane.....	2-23
Power Supply.....	2-24
Job Processor.....	2-24
Memory Boards.....	2-25
Memory Boards in Multibus Systems.....	2-25
Memory Boards in VMEbus-Based Systems.....	2-25
Memory Board Overview .....	2-25
Communication Processors.....	2-26
Intelligent Communications Processor.....	2-28
Advanced Communications Processor .....	2-28
VMEbus Communications Processor.....	2-29
Additional Information.....	2-29
Disk and Tape Controllers.....	2-30
Intelligent Mass Storage Processor (IMSP) .....	2-31
Enhanced Mass Storage Processor (EMSP) .....	2-32
AMSP Disk Controller.....	2-32
SDC-Fixed ESDI Disk Controller — P/90 .....	2-33
1/4-Inch Cartridge Tape Controller — P/90 .....	2-33
1/2-Inch Cartridge Tape Controller — P/90 .....	2-33
1/4-Inch Cartridge Tape Controller — P/95 .....	2-33
Reel-to-Reel Tape Controller — P/60 and P/75 .....	2-33
Reel-to-Reel Tape Controller — P/95 .....	2-33
Disk Drives.....	2-34
Optical Disk .....	2-34
Tape Drives .....	2-35
Tape Drive Features .....	2-35
Storing Data on Tape .....	2-37
Peripheral Devices .....	2-38
Terminals.....	2-38
Modems.....	2-38
Line Printers .....	2-39
Parallel Printers.....	2-39
Serial Printers.....	2-39
 SOFTWARE COMPONENTS.....	 3-1
Sys5 Operating System Overview.....	3-2
File Types.....	3-4
Sys5 Root File System Structure .....	3-5
Root File System Description.....	3-6
bin directory.....	3-8
dev directory .....	3-8
Major Device Numbers.....	3-9
Major Block Device Numbers (Multibus).....	3-10
Major Character Device Numbers (Multibus).....	3-11
Major Block Device Numbers (VMEbus).....	3-14
Major Character Device Numbers (VMEbus) .....	3-15

Inside conf.c .....	3-18
etc directory .....	3-22
usr directory .....	3-27
stand directory .....	3-29
When to Run Standalone Programs .....	3-32
STARTUP AND SHUTDOWN .....	4-1
Connecting the System Console .....	4-2
Normal Startup .....	4-2
PROM Self-test and Diagnostics .....	4-2
Startup, Login, and Shutdown Procedures .....	4-3
Logging In .....	4-6
Setting Up System Passwords .....	4-7
Logging Out .....	4-7
Shutting Down the System .....	4-8
System States .....	4-10
Entering Single-User Mode .....	4-10
Entering Multi-User Mode .....	4-11
Walk-through of /etc/rc/ File .....	4-22
GETTING STARTED .....	5-1
Setting Up Root's Login Environment .....	5-2
System States .....	5-6
Finding the Current State .....	5-6
Changing From State 2 to State s .....	5-7
Changing From State s to State 2 .....	5-7
Additional System Modes .....	5-7
Auto Boot .....	5-9
Modifying /etc/rc .....	5-11
Multibus Systems With ACPs and/or ICPs .....	5-11
VMEbus-Based Systems With VCPs .....	5-13
Sample /etc/rc/ for a VMEbus System .....	5-14
Modifying Maximum File Size .....	5-14
Accounting .....	5-17
Procedure to Enable System Activity Reports .....	5-17
Assigning Holidays and Prime Time .....	5-18
Using Tape Drives .....	5-19
Using the Cartridge Tape Drive .....	5-19
Basic Operation .....	5-21
Features .....	5-22
Retensioning .....	5-22
Using the Reel-to-Reel Tape Drives .....	5-23
Tape Drive Controls .....	5-26
Tape Drive Modes .....	5-27
Improving Tape Drive Performance .....	5-27
Automatic Loading .....	5-28
Automatic Unloading .....	5-29
Manual Loading .....	5-29
Manual Unloading .....	5-32
Reel-to-Reel Tape Drive Error Messages .....	5-32
Cipher 990 Error Messages .....	5-33

Cipher F880 Error Messages.....	5-35
Cipher M891 Error Messages.....	5-37
Making Backups.....	5-41
SETTING UP I/O DEVICES.....	6-1
Creating Nodes in /dev and Building /etc/inittab .....	6-2
Setting Up Terminals.....	6-3
ICP I/O Port Connections.....	6-5
ICP Serial I/O Port Connections.....	6-5
ICP Parallel Connections.....	6-7
ACP I/O Port Connections.....	6-8
ACP Serial I/O Port Connections.....	6-8
ACP Parallel Printer Port Connections .....	6-11
VCP/Link I/O Port Connections.....	6-12
P-4 Link Card I/O Port Connections.....	6-14
Configuring a VCP.....	6-15
Loading an /etc/vconfig File .....	6-17
Connecting a Parallel Printer.....	6-18
Connecting a Serial Printer .....	6-22
Setting Up Modems.....	6-28
Setting Up an Autodial Modem .....	6-28
Setting Up a Dial In Modem .....	6-30
OVERVIEW OF SYS5 FILE SYSTEMS.....	7-1
Physical and Logical Disk Overview .....	7-2
Systems With IMSP and EMSP Controllers .....	7-3
File System Organization .....	7-4
Boot Block.....	7-6
Superblock.....	7-6
i-list.....	7-6
Data Area.....	7-7
Indirect Blocks.....	7-7
Free Space.....	7-8
File Systems — a Closer Look .....	7-9
File Naming Conventions.....	7-9
Linking Files.....	7-10
Copying Files .....	7-11
Moving Files.....	7-12
Additional File Systems.....	7-13
Accessibility.....	7-13
Backup Schemes.....	7-14
Protection.....	7-14
Disk Capacity.....	7-14
Mounting File Systems .....	7-15
Mounting Considerations.....	7-16
File System Status.....	7-19
Using the Ramdisk Driver .....	7-20

CONFIGURING DISKS .....	8-1
Introducing dconfig, mkfs, and mknod .....	8-2
Using dconfig.....	8-2
Using mkfs .....	8-4
Using mknod.....	8-5
Creating Additional File Systems .....	8-6
Calculating Total Sectors Available .....	8-8
Cylinder Size.....	8-12
Step-by-Step Disk Configuration Overview.....	8-12
Step-by-Step Disk Configuration Example.....	8-14
Configuring a Second Disk Drive .....	8-30
CHANGING ROOT AND SWAP .....	9-1
Before You Begin .....	9-2
Root File System Size .....	9-2
An Overview of the Swap Area.....	9-2
Determining the Size of the Swap Area .....	9-3
Increasing, Decreasing, or Moving the Swap Area.....	9-5
Increasing the Root File System and the Swap Area .....	9-9
Example 1.....	9-9
Example 2 .....	9-19
Example 3 .....	9-27
CREATING USER ACCOUNTS.....	10-1
Procedure to Add a User .....	10-3
Creating login Directories.....	10-3
Creating /etc/passwd Entries.....	10-4
Creating /etc/group Entries.....	10-6
Limitations on the /etc/group File.....	10-7
Creating .profile or .cshrc.....	10-8
Setting the Path (Bourne Shell).....	10-10
Setting the Path (C Shell) .....	10-12
Changing Individual and Group Ownerships .....	10-12
Changing Group Affiliations.....	10-14
Setting Terminal Type .....	10-15
Troubleshooting Guide .....	10-16

BACKUP PROCEDURES.....	11-1
Types of Backups.....	11-2
When to Make Backups.....	11-6
Utilities to Backup File Systems.....	11-7
Using fbackup.....	11-7
Using volcopy.....	11-8
Using dump for Backups.....	11-10
Scheduling Backups.....	11-10
Procedure to Take a Full Dump.....	11-11
Procedure to Make Incremental Dumps.....	11-13
Using restor.....	11-14
Using /etc/restor.....	11-14
Using Standalone restor.....	11-15
Restoring an Individual File.....	11-16
Utilities to Back Up Individual Files.....	11-19
Using cpio.....	11-19
Copying Files to Tape With cpio.....	11-19
Copying Files From Tape to Disk With cpio.....	11-20
Automating your Backup Procedures.....	11-21
Using tar to archive.....	11-23
Using find, findc, and frec.....	11-25
Using dd.....	11-26
DISK MIRRORING.....	12-1
Disk Mirroring Overview.....	12-1
Disk Mirroring Requirements and Limitations.....	12-2
Example of Disk Mirroring.....	12-3
Using the mirutil Utility.....	12-4
Invoking the mirutil utility (with options).....	12-4
Suggestions on How to Use mirutil.....	12-5
Using the Mirror Disk Menu.....	12-6
Mirutil Utility Options.....	12-9
Status of Disks on the System.....	12-9
Mirror a Pair of Disks.....	12-11
Re-Connect A Disconnected Disk.....	12-14
Disconnecting a Mirrored Pair.....	12-16
Un-mirror a Pair of Disks.....	12-19
Verify a Mirror Pair.....	12-21
Quit the Mirror Disk Program.....	12-23
Shutting Down Your System With Mirrored Disks.....	12-24
In Case of Difficulty.....	12-24
System Goes Down While Mirroring Two Drives.....	12-24
Root and Secondary Drives Become Unmirrored.....	12-26

MANAGING SYSTEM PROCESSES .....	13-1
Overview .....	13-2
Process Status Command .....	13-3
What is getty? .....	13-5
More About Process Status .....	13-8
Fork and Execute .....	13-10
Killing Processes .....	13-12
Who Can Kill Processes? .....	13-12
How to Free Up Frozen Terminals .....	13-13
Freeing Up A Terminal .....	13-13
Freeing Up All Terminals and the System Console .....	13-16
Thrashing .....	13-17
CRISIS RECOVERY .....	14-1
Trouble Shooting Guide .....	14-2
Recovering from Ungraceful Shutdowns .....	14-3
Restarting the System .....	14-3
Repairing File Systems .....	14-4
Recovering Lost Data .....	14-5
Running restor .....	14-5
Restoring Files and Directories .....	14-6
Restoring Entire Disk File System .....	14-9
Recording Memory, ICP, ACP, and VCP Dumps .....	14-12
Memory, ICP and ACP Dump Procedure .....	14-12
Memory and ACP Dump Procedure .....	14-16
Contents of Memory and ACP Dump Tapes .....	14-20
Memory and VCP Dump Procedure .....	14-21
Contents of Memory and VCP Dump Tapes .....	14-25
STAND-ALONE PROGRAMS .....	15-1
Overview .....	15-1
Plexus Release Tape Stand-alone Programs .....	15-3
How to Run a Stand-alone Program .....	15-6
The Long Method .....	15-7
Loading Stand-alone Programs from Disk .....	15-7
Loading Stand-alone Programs from Tape .....	15-8
The Short Method .....	15-9
Arguments to Stand-alone Programs .....	15-10
The Mount and Skip Commands .....	15-12
Descriptions of Stand-alone Programs .....	15-13
help .....	15-13
unix .....	15-13
dformat .....	15-14
Executing dformat .....	15-15
How to Use This Program .....	15-19
IMSP Examples .....	15-20
Format (IMSP) Without a Bad Track Tape .....	15-20
With a Bad Track Tape .....	15-22
Spare (IMSP) .....	15-23

Read (IMSP) .....	15-23
List (IMSP) .....	15-23
EMSP Examples .....	15-24
Format (EMSP) Without a Bad Track Tape .....	15-24
With a Bad Track Tape.....	15-24
Spare (EMSP) .....	15-27
Read (EMSP) .....	15-27
List (EMSP) .....	15-27
Verify (EMSP).....	15-28
AMSP Examples .....	15-28
Format (AMSP) Without a Bad Track Tape.....	15-28
With a Bad Track Tape.....	15-31
Spare (AMSP).....	15-32
Read (AMSP).....	15-33
List (AMSP).....	15-33
mkfs.....	15-34
dconfig.....	15-37
fsck.....	15-38
restor.....	15-40
dd .....	15-40
fbackup.....	15-40
Other Stand-alone Programs .....	15-41

## List of Figures

INTRODUCTION.....	1-1
HARDWARE COMPONENTS .....	2-1
Figure 2-1. Devices in the Main Cabinet Controlled by the SCSI.....	2-15
SOFTWARE COMPONENTS.....	3-1
Figure 3-1. Root File System.....	3-5
Figure 3-2. Listing the Contents of the Root Directory.....	3-5
Figure 3-3. Listing the Root File System With the Long Option.....	3-6
Figure 3-4. Partial Listing of the /dev Directory.....	3-9
Figure 3-5. Listing (ls -l) of the /stand Directory (Multibus).....	3-30
STARTUP AND SHUTDOWN .....	4-1
Figure 4-1. Sample /etc/rc File for 68000 (Multibus) Systems .....	4-12
Figure 4-2. Sample /etc/rc File for 68020 (Multibus) Systems .....	4-17
Figure 4-3. Sample /etc/rc File for 68020 (VMEbus) Systems .....	4-28
GETTING STARTED .....	5-1
Figure 5-1. Examples Downloading VCPs .....	5-13
Figure 5-2. Sample VMEbus-Based /etc/rc File.....	5-14
Figure 5-3. Tape Unit, Tape Path.....	5-31
Figure 5-4. Cipher F880 and M891 Diagnostic Indicator Values ..	5-36
SETTING UP I/O DEVICES.....	6-1
Figure 6-1. ASYNC: One VCP and Two LINC Cards.....	6-16
Figure 6-2. PARA: Two VCP's With One LINC Card Each .....	6-16
Figure 6-3. Sample /etc/gettydefs File.....	6-32
OVERVIEW OF SYS5 FILE SYSTEMS.....	7-1
Figure 7-1. mkfs File System Organization.....	7-5
CONFIGURING DISKS .....	8-1
Figure 8-1. Sample Disk Organization (512-byte sectors) .....	8-15
Figure 8-2. dconfig Display, IMSP Drives .....	8-19
Figure 8-3. dconfig Display, EMSP Drive.....	8-19
Figure 8-4. dconfig Display, AMSP Drive.....	8-20
Figure 8-5. dconfig Display, P/90 Drive.....	8-20
Figure 8-6. Sample dconfig Display.....	8-23
Figure 8-7. Inserting mounts and umounts in /etc/rc.....	8-29



CHANGING ROOT AND SWAP .....9-1

CREATING USER ACCOUNTS .....10-1

BACKUP PROCEDURES .....11-1

Figure 11-1. Sample Backup Script .....11-22

DISK MIRRORING .....12-1

Figure 12-1. An Example of Mirrored Disks .....12-3

Figure 12-2. Mirror Disk Menu Display .....12-6

Figure 12-3. Status of Disks on the System .....12-9

Figure 12-4. Explanation of Mirror Disk Status Listing .....12-10

Figure 12-5. Connecting a Pair of Mirrored Disks .....12-12

Figure 12-6. Status After Connecting a Mirrored Pair .....12-13

Figure 12-7. Re-attaching a Detached Disk .....12-15

Figure 12-8. Disconnecting a Mirrored Pair .....12-18

Figure 12-9. Un-mirroring a Mirrored Pair .....12-20

Figure 12-10. Mirror Status After Un-mirroring a Disk .....12-20

Figure 12-11. Status While the Verify Operation is Taking Place..12-22

Figure 12-12. System Goes Down While Mirroring Disks .....12-25

MANAGING SYSTEM PROCESSES .....13-1

Figure 13-1. A Sample ps Listing .....13-3

Figure 13-2. A Partial Listing Illustrating the ps -ef Command .....13-4

Figure 13-3. Sample Listing of /etc/gettydefs .....13-6

Figure 13-4. Partial Listing of ps -ef .....13-8

Figure 13-5. A Sample Listing Illustrating Process ID Numbers...13-14

CRISIS RECOVERY .....14-1

Figure 14-1. VCP LED Activity Indicators .....14-21

STAND-ALONE PROGRAMS .....15-1

## List of Tables

INTRODUCTION.....	1-1
Table 1-1. Plexus System Description.....	1-3
Table 1-2. Sample Logbook Entries.....	1-8
HARDWARE COMPONENTS .....	2-1
Table 2-1. P/35 Card Cage Slot Assignments .....	2-3
Table 2-2. P/35 Dimensions and Weight .....	2-4
Table 2-1. P/55 Card Cage Slot Assignments .....	2-6
Table 2-2. P/60 Card Cage Slot Assignments .....	2-8
Table 2-3. P/75 Card Cage Slot Assignments .....	2-11
Table 2-4. P/90 Card Cage Slot Assignments .....	2-16
Table 2-5. P/95 Card Cage Layout (Stand-alone) .....	2-18
Table 2-6. P/95 Card Cage Layout (One Expansion Cabinet) .....	2-19
Table 2-7. P/95 Card Cage Layout (Two Expansion Cabinets) .....	2-20
Table 2-8. System I/O Bus Summary .....	2-23
Table 2-9. Job Processors in Plexus Systems.....	2-24
Table 2-10. Communication Processors in Plexus Systems .....	2-26
Table 2-11. Number of Serial I/O Ports — Multibus Systems.....	2-27
Table 2-12. Disk and Tape Controllers Available .....	2-30
Table 2-13. Tape Drives in Plexus Systems.....	2-35
SOFTWARE COMPONENTS .....	3-1
STARTUP AND SHUTDOWN.....	4-1
Table 4-1. Console Port Pinouts .....	4-2
GETTING STARTED .....	5-1
Table 5-1. System States .....	5-6
Table 5-2. System Modes.....	5-8
Table 5-3. Cartridge Tape Device Names.....	5-19
Table 5-4. P/35, P/55, and P/75 Cartridge Tape Features.....	5-22
Table 5-5. P/60 and P/75 Reel-to-Reel Tape Drives.....	5-23
Table 5-6. Reel-To-Reel Tape Device Names .....	5-23
Table 5-7. Tape Controls Used With Reel-to-Reel Tape .....	5-24
Table 5-8. Cipher F880 Error Conditions During Tape Load .....	5-35
Table 5-9. Cipher M891 Error Messages .....	5-38

SETTING UP I/O DEVICES.....6-1

Table 6-1. ICP Serial Port Pinout.....6-5

Table 6-2. ICP Terminal or Serial Printer Cable Pinouts .....6-5

Table 6-3. ICP Asynchronous Modem Cable Pinouts.....6-6

Table 6-4. ICP Synchronous Modem Cable Pinouts.....6-6

Table 6-5. ICP Parallel Printer Pinout Connections .....6-7

Table 6-6. ACP Terminal/Serial Printer Cable Pinouts.....6-8

Table 6-7. ACP Asynchronous Modem Cable Pinouts .....6-9

Table 6-8. ACP Synchronous Modem Cable Pinouts.....6-9

Table 6-9. Data Carrier Detect and Clear to Send Signals.....6-10

Table 6-10. ACP Parallel Printer Pinout Connections.....6-11

Table 6-11. Terminal/Serial Printer Cable Pinouts (DB-9).....6-13

Table 6-12. Synchronous Modem Cable Pinouts (DB-9).....6-13

Table 6-13. Asynchronous Modem Cable Pinouts (DB-9).....6-13

Table 6-14. Parallel I/O Port Pinouts (P-4 Link) .....6-14

Table 6-15. Link ID Numbers .....6-15

OVERVIEW OF SYS5 FILE SYSTEMS.....7-1

CONFIGURING DISKS .....8-1

Table 8-1. Sample Dconfig Display in 512 Byte Sectors.....8-7

Table 8-2. Total Sectors and Blocks Available.....8-8

Table 8-3. AMSP Disk Configuration Values .....8-10

Table 8-4. P/90 Disk Configuration Values .....8-10

Table 8-5. IMSP Disk Configuration Values.....8-10

Table 8-6. EMSP Disk Configuration Values .....8-11

Table 8-7. Minor Device Numbers Used in mknod .....8-25

Table 8-8. Software Interleave and Blocks per Cylinder.....8-32

CHANGING ROOT AND SWAP .....9-1

Table 9-1. Example 1 — Disk Configuration .....9-9

Table 9-2. Example 1 — File System Size .....9-10

Table 9-3. Software Interleave and Blocks per Cylinder.....9-15

Table 9-4. Example 2 — Disk Configuration .....9-19

Table 9-5. Example 2 — File System Size .....9-20

Table 9-6. Example 3 — Disk Configuration .....9-27

Table 9-7. Example 3 — File System Size .....9-28

**C**

CREATING USER ACCOUNTS.....10-1

Table 10-1. Command Summary .....10-2

Table 10-2. Troubleshooting Guide .....10-17

BACKUP PROCEDURES.....11-1

Table 11-1. Types of Backup Utilities .....11-2

Table 11-2. Sys5 Programs that Write to Tape .....11-4

DISK MIRRORING .....12-1

MANAGING SYSTEM PROCESSES.....13-1

CRISIS RECOVERY .....14-1

Table 14-1. Different Methods of Making Dump Tapes.....14-9

Table 14-2. ACP LED Status Indications .....14-17

STANDALONE PROGRAMS .....15-1

Table 15-1. Total Number of Sectors Created by dformat .....15-18

Table 15-2. Software Interleave and Blocks per Cylinder.....15-36





# INTRODUCTION

---

This chapter:

- introduces you to the Plexus systems this manual is written for.
- explains how to use this manual.
- provides a checklist of procedures to complete.
- defines who should read this manual.
- explains the responsibilities of the system administrator, including how to maintain a system logbook.
- describes the major types of users and how to become the superuser.

## Before You Begin

Before you begin, read the Preface. Not only does the Preface describe how this manual is organized, but it also:

- refers you to a guide, *Read Me First*, which outlines the contents of Plexus and A T & T manuals.
- provides an annotated bibliography of commercially available UNIX books.

---

## Introduction to Plexus Systems

This manual describes how to perform Sys5 system administration procedures on Plexus P/35, P/55, P/60, P/75, P/90 and P/95 systems.

Plexus' newest system, the P/90, features a powerful, true 32-bit 68020 job processor. The surrounding circuitry allows it to execute at clock speeds up to 20MHz while maintaining zero wait states. The P/90 system also features the industry standard VMEbus, a 1/4-inch 60Mb cartridge tape drive, and one VMEbus communication processor (VCP). It runs Plexus' Sys5 version of the UNIX operating system.

The P/90 system is expandable and can provide up to:

- 32 serial ports (with 4 DB-9 Link cards)
- 4 parallel ports (with 1 P-4 Link card)
- 16Mb of main memory
- three disk drives in the main cabinet and up to sixteen disk drives in a fully expanded P/90 (main cabinet plus two expansion cabinets)
- two expansion cabinets (each housing up to 8 additional disk drives and two types of removable media) (i.e. 1/4-inch cartridge tape, 1/2-inch cartridge tape (expected to be available at a later date), and optical disk (expected to be available at a later date)).

The P/90 is object-code compatible with the full-line of Plexus 68020-based systems, making it easy to transport applications from one Plexus system to another.

This manual refers to Plexus systems in two major categories: Multibus systems and VMEbus systems. The following table defines Plexus Multibus and VMEbus systems. Refer to this table as you read the remainder of this manual.

---

<b>Table 1-1. Plexus System Description</b>	
<b>Multibus Systems</b>	<b>VMEbus Systems</b>
P/35, P/55, P/60, P/75	P/90, P/95

## How to Use This Manual

Read this manual chapter by chapter. That is, start with this chapter, and continue to Chapter 2. Next, read Chapter 3 and so on. Do not skip chapters unless this manual tells you to do so. The chapters are organized in a step-by-step fashion.

See the *Sys5 UNIX Error Message Reference Manual* for a complete description of error messages.

## Checklist to Complete

This manual assumes that you have installed the Plexus system using the *Installation Guide* that shipped with your Plexus system. A checklist of procedures to configure and maintain the Sys5 system software is provided below. Check off each item in the checklist as you complete the procedure.

- |                          |                                 |                    |
|--------------------------|---------------------------------|--------------------|
| <input type="checkbox"/> | Introduces Plexus systems       | Preface, Chapter 1 |
| <input type="checkbox"/> | Describes hardware and software | Chapters 2-3       |
| <input type="checkbox"/> | Startup the System              | Chapter 4          |
| <input type="checkbox"/> | Set Up Root's Login Environment | Chapter 5          |
| <input type="checkbox"/> | Set Up I/O Devices              | Chapter 6          |
| <input type="checkbox"/> | Learn about Sys5 File Systems   | Chapter 7          |
| <input type="checkbox"/> | Increase Root and/or Swap       | Chapter 8          |
| <input type="checkbox"/> | Configure Disks                 | Chapter 9          |
| <input type="checkbox"/> | Create User Accounts            | Chapter 10         |



---

At this point, your Plexus system is set up and ready to use. You will need to perform tape backups, preventive maintenance procedures, and check the system's disk free space. You will also need to reboot the system and check the file systems when:

- the system is brought down gracefully
- and if the system crashes.

The following chapters contain information pertaining to day-to-day operations.

Shutdown the system	Chapter 4
Backup Procedures	Chapter 11
Disk Mirroring	Chapter 12
Managing System Processes	Chapter 13
Crisis Recovery	Chapter 14
Stand-alone Programs	Chapter 15

## Duties of the System Administrator

One person must be appointed to take overall responsibility to ensure the soundness of your Plexus system and its Sys5 operating system. This person is the system administrator. This document, which explains how to configure and set up your Plexus system, is written for a system administrator with a UNIX background.

At times, the day-to-day management of a multi-user system can be unpredictable and exciting. However, there are certain routines that must be followed. Some of the routines a system administrator performs include:

- starting up and shutting down the system
- configuring disks
- setting up file systems
- setting up serial and parallel devices (terminals, printers, and modems)

- 
- setting up the printer spooler
  - setting up the root account (**.profile** for the Bourne shell or **.cshrc** and **.login** for the C shell).
  - adding new users to the system and setting up users' environments
  - removing obsolete users and reorganizing their files
  - making tape backups of file system(s)
  - restoring file systems and files
  - monitoring systems operation and resources (disk free space, file management, etc.)
  - performing preventive maintenance (cleaning tape drive heads, cleaning filters).
  - installing and maintaining application software
  - performing recovery if the system crashes
  - solving users' problems
  - installing new software releases
  - maintaining the system logbook

The system administrator will also be the point of contact for user questions and problems. Effectively answering questions and solving problems affects how the system operates. It is extremely beneficial to have an in house expert available should users have questions about the system or Sys5. Problem solving skills are required for the system administrator to narrow down, diagnose, research, and solve a system software problem which might affect users.

---

## Alternate System Administrator

There should also be a designated alternate to the system administrator. This person should be just as knowledgeable as the primary administrator. The system administrator and alternate system administrator should cooperate with each other to ensure adequate coverage and supervision of the system.

**[ NOTE ]** It is recommended that there be at least one primary and one alternate system administrator for each Plexus system installed.

Both system administrators should work together so that both acquire extensive experience and practice in working with the system. Most of the system administrator's responsibilities are actually fairly easy to perform. Once the system is set up and has stabilized, the system administrator will find that the actual duties to be performed will take less and less time.

During the initial setup, however, the duties may take a lot of time. This is because the task of configuring the system is coupled with the system administrator's lack of experience.

It is also recommended that during the initial familiarization stage, the system be used as a learning tool for any and all assigned to the system before the system is brought up and made generally available to users. This is usually a difficult task to work out with impatient managers who see the system being used for playing and not for constructive work. But, if the system administrator is new to the Sys5/Plexus environment, the time spent in privately learning the system at the outset will pay off later in experienced system management with a minimum of corruption to important work files.

---

## System Logbook

The system logbook is a critical tool used by the system administrator to organize the above responsibilities. The system administrator's running record of the total system activity should, at an absolute minimum, include the following:

1. The date your Plexus system was delivered, installed, and accepted.
2. Detailed dated records on how the system was configured, including:
  - a. file systems created, their names, sizes, labels, etc.
  - b. user accounts which were added or deleted.
  - c. information explaining which system ports are connected to which peripherals (i.e. terminals, printers, etc.)
  - d. installation of any application software (where installed).
  - e. any changes made to any of the files in the root file system.
3. Dated reports of system failure including a detailed listing of any error messages displayed on the console terminal as well as any information which can be obtained on system activity prior to the failure. This is critical!
4. Dated reports on any visits by Plexus personnel and services performed (e.g. what components were replaced, what software was modified or recreated).
5. The dates of any file system checking errors encountered and what was done to correct such problems.
6. The dates of file system backups, including the file system name and what procedure was used to back up file systems.

---

Table 1-2 presents an example of logbook entries.

**Table 1-2. Sample Logbook Entries**

<u>DATE</u>	<u>TIME</u>	<u>PROCEDURE</u>
6/16/86	1100:	Full backups of <b>/dev/dsk/0s1</b> (root) using <b>/etc/dump</b> ; full backups of <b>/dev/dsk/0s3</b> (/user) and <b>/dev/dsk/0s4</b> (/lang) using <b>cpio</b> .
6/16/86	1600:	Installed Rev 1.4; increased Nswap to 20000 Installed <b>cpio</b> of <b>/etc/passwd</b> , <b>/etc/profile</b>
6/17/86	0800:	Added user <b>jones</b> , group <b>admin</b> mode <b>775</b> ; Added user <b>ryan</b> , group <b>mktg</b> mode <b>664</b>
6/17/86	1200:	Incremental backup of <b>/dev/dsk/0s3</b> (/user) and <b>/dev/dsk/0s4</b> (/lang) using <b>cpio</b> .

## Types of Users

On a Sys5 system there are two types of users:

- Privileged users
- Functional users

Briefly, privileged users are responsible for system administration of a Sys5 system and functional users are normal users. A discussion of both types of users follows.

## Privileged Users

Privileged users have special powers and privileges beyond those of ordinary users. Besides root and bin, there are other privileged users including adm (for system accounting), lp (for spooler administration), and sys (for system files).

---

The first privileged user is root (the super-user). When you log into the system as root and give the appropriate password, you can execute any system command, and access any directory or file, regardless of protections. Virtually all procedures for system configuration and setup require that you be logged in as root.

The second and less used user identity is bin. The bin user does not have special privileges directly. However, bin does own many important system files. This ownership allows the bin user to do many things to these files. Beyond widespread directory/file ownership, including all system commands, bin has no special powers.

When you log into the system as bin, you are placed in bin's home directory, **/bin**. This directory contains most of the Sys5 command files, most of which are owned by bin. Some bin files are owned by root because they are set user id (setuid) programs and will not run properly if ownership is changed to bin.

## How to Become the Root Super-user

There are three (3) ways to become the super-user.

1. At the login prompt, enter:

**root**

then enter the correct password. The system will place you in root's home directory ( / ) . Your shell prompt (#) will remind you that you possess super-user powers.

2. When you are logged into the system as yourself you can take transitive powers of root without logging off by entering:

**su**

and the proper root password. This "switch user" command is frequently called going super-user. It spawns a super-user shell for you to work with until you enter:

**<CTRL d>**

to kill that special process, and return to your normal user state. When you go super-user, you are kept working in your current directory, and returned there when you exit. A log of **su** activity is kept in the file **/usr/adm/sulog**.

---

To obtain the true root environment use the `-` option with `su`:

`su -`

This gives you the true root environment until you enter:

`<CTRL d>`

which kills that special process and returns you to your normal user state.

3. When the system is brought up to the single-user state, whoever is on the console terminal is immediately given super-user powers. The reason for this is that all of the standard procedures performed in single-user mode, such as file system checking, repairing, backing up, mounting, and unmounting require root's powers.

### **Powers of the Root Super-User**

**[ NOTE ]** Logging in as either administrative user gives you broad powers, which can be easily misused. Be aware of what directory you are in when you create trash/working files. DO NOT corrupt the integrity of important system directories such as: `root ( / )`, `/bin`, or `/etc`. Also, do not corrupt the integrity of the Plexus released file system `/dev/dsk/0s1`.

Permissions and restrictions do not apply to the root user. Therefore, root can read and/or write anyone's files/directories or remove them. Hence, if you are logged in as root, you have the potential to destroy anything from a single file up to an entire file system. Without cautious use of these powers, a properly functioning Sys5 file system can be reduced to a critical mess. For instance, by entering an incorrect command, root could remove most of the system commands — thus paralyzing the system.

---

If such a command is executed, the system administrator must:

- halt the system
- restore the entire root file system from a tape backup

This is a time consuming and disruptive task. What makes matters worse is that there are many commands that can allow total dismemberment of an otherwise healthy system.

As a system administrator you must recognize the swift and ubiquitous powers of the super user. Carefully heed important precautions that follow.

**Precaution:**

Keep super-user login activity to an absolute minimum.

**Why:**

By using super-user powers only when absolutely required, you minimize the possibility of opening up Pandora's box. It is not wise for anyone to do every-day system work as the super-user.

**Precaution:**

Limit knowledge of the super-user password.

**Why:**

It would be best if only two individuals, the system administrator and the alternate system administrator, know the root password. Logging in as **root** or **su** is the only key to super-user powers, therefore, limiting the number of system users who can become the super-user limits super-user power.

**Precaution:**

Carefully check each command before pressing RETURN.

**Why:**

The normal Bourne shell prompt is `$` . The normal C shell prompt is `%` . The root prompt is `#` . The `#` prompt should remind you to carefully check each entry before pressing RETURN. There are no restrictions to super-user power; what you enter is carried out.



---

## Functional Users

All other users on Plexus Sys5 systems are referred to as functional users. The functional user accounts are really the workhorse accounts of the system. That is where all the on-the-job files and directories are created, used, and destroyed. Unlike root and bin, each functional user id (UID) and home directory (HOME) must be created by the super-user.

**[ NOTE ]** The home directories of ordinary users are created by the system administrator and should reside on an alternate file system from root and bin. This is to partition storage of user work files from Sys5 system files.

Like bin, the power of normal users extends over the files they own. Their ability to use (i.e. read, write, or execute) other users' files depends on the protections assigned to those files by the other users or the super-user.

## System Passwords

Your Plexus system comes with no password established for root, bin, and other privileged users. Therefore, it is wise to establish passwords for all predefined accounts as soon as the system is in a booted state. Root can establish a password for itself and anyone else. See the section "Setting Up System Passwords" in the chapter *Startup and Shutdown* for additional information.

# HARDWARE COMPONENTS

---

## Overview

This chapter describes the basic hardware components of your system. It gives you background information about different hardware categories.

The major topics covered are:

- hardware components comprising your P/35, P/55, P/60, P/75, P/90, or P/95 Plexus system
- individual system descriptions
- peripheral devices available

---

## **Overview of Plexus Systems**

This section describes the basic system configuration, board placement, and expansion capabilities of the P/35, P/55, P/60, P/75, P/90, and P/95 systems.

### **P/35 System**

This section describes:

- component parts of the P/35
- slot assignments
- controls and indicators
- dimensions and weight
- expansion capabilities of the P/35

### **P/35 System Description**

The basic P/35 system contains:

- 68000 job processor (can be upgraded to 68020)
- Up to 8Mb of memory (two 4Mb memory boards)
- Up to two Intelligent Communication Processors (ICPs) (can be upgraded to two Advanced Communication Processors)
- Hard disk/cartridge tape controller (IMSP)
- Power supply
- Multibus backplane
- One hard disk drive
- 45Mb cartridge tape drive

The second memory board and second ICP are optional.

---

## P/35 Slot Assignments

The P/35 card cage slot assignments are presented in Table 2-1. Note that second memory (Memory 1) and second ICP/ACP (ICP/ACP 1) are optional.

**Table 2-1. P/35 Card Cage Slot Assignments**

Slot #	Non-Ethernet	Ethernet	Ethernet
	Board Name	Config 1 Board Name	Config 2 Board Name
1	Memory 0	Memory 0	Memory 0
2	Memory 1	Ethernet	Memory 1
3	Processor	Processor	Processor
4	ICP/ACP 1	ICP/ACP 1	Ethernet
5	ICP/ACP 0	ICP/ACP 0	ICP/ACP 0
6	IMSP	IMSP	IMSP

Note that Ethernet is positioned in slot 2 with one memory board. If your system has one communications processor, the Ethernet board is positioned in slot 4.

## P/35 Controls and Indicators

The P/35 system controls are:

- a SYSTEM POWER on/off keyswitch and a power on indicator which are mounted on the front panel of the unit.
- a RESET pushbutton which is mounted on the rear panel of the unit. The RESET pushbutton, when pushed, resets the processor and reboots the system software.

## P/35 Dimensions and Weight

The outside dimensions of the P/35 are presented in the following table.

---

**Table 2-2. P/35 Dimensions and Weight**

<b>Dimension</b>	<b>English</b>	<b>Metric</b>
Height	10.5 in.	27 cm.
Width	19 in.	48 cm.
Depth	24 in.	61 cm.
Weight	65 lb.	29 kg.
Shipping weight	75 lb.	34 kg.

### **P/35 Storage Expansion Module**

A custom designed System Expansion Module (SEM) is available. This unit is a small cantilever stand designed to hold a single P/35 system. The SEM expands the P/35 disk capacity to three disk units (one internal, two external). The SEM consists of a top, a column, and a base. The column can house two 8-inch disk units and their associated cables; the base can house the disk drive power supply units. Instructions for mounting a P/35 system onto a SEM are given in the *SEM Installation Guide* manual.

---

## **P/55 System**

This section describes:

- component parts of the P/55
- P/55 slot assignments
- expansion capabilities of the P/55
- additional information

### **P/55 System Description**

The basic P/55 system contains:

- 68020 job processor
- Up to 8Mb of memory (two 4Mb memory boards)
- Up to two Advanced Communication Processors (ACPs)
- Hard disk/cartridge tape controller (IMSP)
- Power supply and cooling fans
- Multibus backplane
- One hard disk drive
- 60Mb cartridge tape drive
- One reel-to-reel tape drive
- CPC tape controller

### **P/55 Slot Assignments**

The P/55 slot assignments are indicated in the following table.

---

**Table 2-3. P/55 Card Cage Slot Assignments**

<u>Slot #</u>	<u>Board Name</u>
1	Memory 0
2	Memory 1
3	68020 CPU
4	ACP 1
5	ACP 0
6	IMSP

**[ NOTE ]** If you have an Ethernet board, insert the Ethernet board in either slot 2 (if you have one memory board) or in slot 4 (if you have only one ACP). The second memory board and second ACP are optional.

### **P/55 Storage Expansion Module**

A custom designed System Expansion Module (SEM) is available. This unit is a small cantilever stand designed to hold a single P/55 system. The SEM expands the P/55 disk capacity to three disk units (one internal, two external). The SEM consists of a top, a column, and a base. The column can house two 8-inch disk units and their associated cables; the base can house the disk drive power supply units. Instructions for mounting a P/55 system onto a SEM are given in the *SEM Installation Guide* manual.

### **Additional Information**

For additional information on the P/55's features, see the *P/55 Installation Guide*. The *P/55 Maintenance Manual* provides additional technical information, cabling requirements, and preventive maintenance procedures.

---

## **P/60 System**

This section describes:

- component parts of the P/60
- P/60 slot assignments
- expansion capabilities of the P/60
- additional information

### **P/60 System Description**

The basic P/60 system contains:

- 68020 job processor (68020 job processor in older systems)
- 1Mb to 16Mb of memory (four 4Mb memory boards)
- up to 5 ICPs (I/O controllers) (can be upgraded to 5 ACPs)
- Hard disk/cartridge tape controller (IMSP)
- Power supply
- Multibus backplane
- One disk drive
- reel-to-reel tape drive with CPC tape controller

### **P/60 Slot Assignments**

The card cage PC board slot assignments for the P/60 are shown in the following table:



---

**Table 2-4. P/60 Card Cage Slot Assignments**

---

<b>Non-Ethernet</b>		<b>Ethernet</b>	
<b>Slot #</b>	<b>Board</b>	<b>Slot #</b>	<b>Board</b>
1	Processor	1	Processor
2	Memory 3	2	Memory 3
3	Memory 2	3	Memory 2
4	Memory 1	4	Memory 1
5	Memory 0	5	Memory 0
6	ICP/ACP 4	6	ICP/ACP 3
7	ICP/ACP 3	7	ICP/ACP 2
8	ICP/ACP 2	8	ICP/ACP 1
9	ICP/ACP 1	9	ICP/ACP 0
10	ICP/ACP 0	10	Tape Controller
11	Tape Controller	11	Ethernet Controller
12	Disk Controller	12	Disk Controller

---

In the previous table note that slot assignments 1 through 5 are reserved for processor and memory only. Slots 6 through 12 can be a combination of input/output controllers: ACP, ICP, disk, tape, ethernet, others, etc.

---

## **P/60 Expansion Cabinet**

An expansion cabinet is available for the P/60 system. The P/60 expansion cabinet is free-standing and located next to the main P/60 system cabinet. The cabinet is mounted on casters. Shielded cables from the expansion cabinet connect to mating connectors inside the P/60 system cabinet.

The P/60 expansion cabinet can contain:

- one reel-to-reel streaming tape drive.
- One or two 443Mb fixed disk drives, with or without the streaming tape drive.
- One or two 586Mb fixed disks.

Space must be left at the rear of the unit to permit opening of the rear cover and access to the rear of the cabinet. Space must also be left in front of the cabinet to enable technicians to extend the slide-mounted units out from the cabinet and to work around these units. In some cases the unit must be pulled out to enable the side cover(s) to be removed and to also extend a unit so that it can be worked on from the side. To satisfy the above conditions it would be necessary to leave a space at least 3 feet in depth behind the cabinet and a space at least 5.5 feet in front of the system.

For additional information, see the *P/60 Expansion Cabinet Installation Guide* manual.

### **Additional Information**

For additional information on the P/60's features, see the *P/60 Installation Guide*. The *P/60 Maintenance Manual* provides additional technical information, cabling requirements, and preventive maintenance procedures.

---

## **P/75 System**

This section describes:

- component parts of the P/75
- expansion capabilities of the P/75
- Common Circuits Board capabilities
- additional information

### **P/75 System Description**

A minimum, single-cabinet P/75 system is composed of the following major components:

- 68020 processor board
- 4Mb (1Mb in older P/75s) to 16Mb of memory
- Advanced Communications Processor I/O Controller (ACP)
- Hard disk/cartridge tape controller (IMSP)
- Power supply and cooling fans
- Multibus backplane
- One or two 8-inch hard disk drives
  - 142Mb drives
  - 286Mb drives
- 60Mb cartridge tape drive

### **P/75 Slot Assignments**

The card cage slot assignments for the P/75 are shown in the following table.

---

**Table 2-5. P/75 Card Cage Slot Assignments**

---

<b>Ethernet</b>		<b>Non-Ethernet</b>	
<b>Slot #</b>	<b>Board Name</b>	<b>Slot #</b>	<b>Board Name</b>
0	CCB	0	CCB
1	68020 CPU	1	68020 CPU
2	Mem 3	2	Mem 3
3	Mem 2	3	Mem 2
4	Mem 1	4	Mem 1
5	Mem 0	5	Mem 0
6	ACP 0	6	ACP 0
7	ACP 1	7	ACP 1
8	ACP 2	8	ACP 2
9	Ethernet	9	ACP 3
10	IMSP	10	IMSP
11	EMSP 2	11	EMSP 2
12	EMSP 1	12	EMSP 1
13	EMSP 0	13	EMSP 0
14	reel-to-reel tape controller	14	reel-to-reel tape controller

---

\* Slots 6-10 can contain up to 5 ACPs.

**[ NOTE ]** When possible, leave empty slots on the component side of the CPU, ACPs, Ethernet, IMSP, and EMSPs.

### **P/75 Expansion Cabinets**

Expansion Cabinets are available for the P/75 computer. The cabinets are mounted on casters.

The P/75 can accommodate up to four expansion cabinets. Each P/75 expansion cabinet bolts to the adjoining P/75 cabinet. Cables within each expansion cabinet connect to an adjoining cabinet through the use of connectors. These connectors are mounted on a panel in a duct connecting the cabinets.

---

Each expansion cabinet can contain the following:

- reel-to-reel streaming tape drive (no disk drives)
- reel-to-reel streaming tape drive plus
  - a. one 443Mb disk drive or
  - b. one 586Mb disk drive or
  - c. two 443Mb disk drives
- One to four 443Mb fixed disk drives (no tape).
- One or two 586Mb fixed disk drives (no tape).

For additional information, see the *Expansion Cabinet Installation Guide* manual.

#### **P/75 Common Circuits Board**

The Common Circuits Board (CCB), which resides in slot 0, performs the following functions:

- Bus clock signal generation
- Priority coding/decoding
- Reset signal generation
- Power control for the P/75 and Expansion cabinets
- Power fail signal generation
- Special Interrupt notification
- Multibus active slot indication
- Two levels of temperature sensing: 45° C ±5°, and 70° C ±5°.
- D.C. voltage level monitoring
- Uninterruptable Power Source Interrupt connection
- On board 300/1200 bps modem (North American systems only)
- LED board status indicators

Note that the built-in auto-dial / auto answer modem is a standard P/75 feature facilitating isolation of hardware and software

---

problems (North American systems only).

For additional information on the CCB, including LED indicator values, see the *P/75 Maintenance Manual*.

**Additional Information**

For additional information on features of the P/75, see the *P/75 Installation Guide*. The *P/75 Maintenance Manual* provides additional technical information, cabling requirements, and preventive maintenance procedures.

---

## **P/90 System**

This section describes:

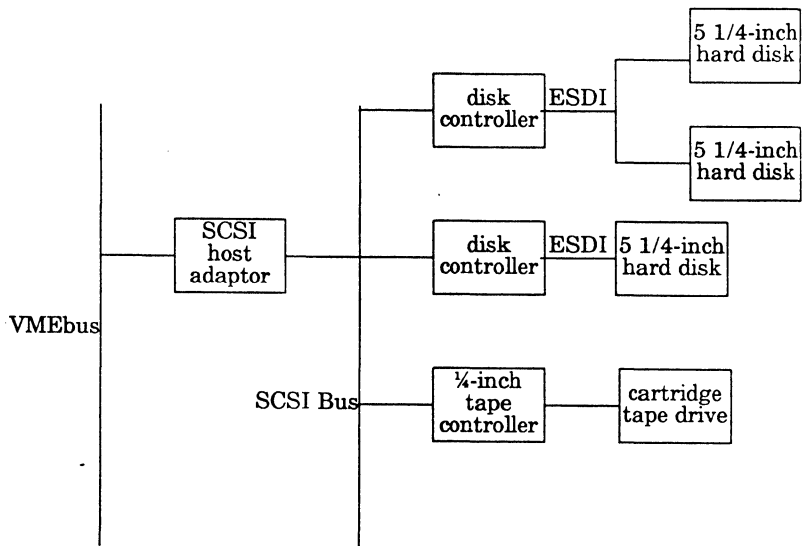
- component parts of the P/90
- P/90 card cage slot assignments
- expansion capabilities of the P/90
- additional information about the P/90 computer system

### **P/90 System Description**

The P/90, Plexus' newest system, is based on 5 1/4-inch disk and tape technology. A minimum single-cabinet P/90 system is comprised of the following components:

- 68020 processor board (20MHz and soon 25MHz)
- One memory board
- One VMEbus Communications Processor I/O Controller (VCP)
- One SCSI host adapter controlling:
  - up to three 5 1/4-inch hard disk drives selected from the following (Fujitsu 67Mb, Fujitsu 135Mb, Fujitsu 315Mb, or Maxtor 260Mb — formatted) using the Enhanced Small Device Interface (ESDI).
  - one 1/4-inch 60Mb cartridge tape.
- Power supply

Figure 2-1 illustrates the disk and tape devices the SCSI host adapter controls.



**Figure 2-1.** Devices in the Main Cabinet Controlled by the SCSI

Each SCSI host adaptor can control up to the following:

- two disk controllers (each controlling up to two hard disks).
- one 1/4-inch cartridge tape controller (controlling one tape drive).
- one 1/2-inch cartridge tape controller (controlling one tape drive). Half-inch cartridge tape is expected to be available at a later date.
- one optical disk controller (controlling 5 1/4-inch optical disk which is expected to be available at a later date).

### **P/90 Slot Assignments**

The memory board, job processor board, and bus arbiter board occupy the first three slots of the card cage (see Table 2-6). A single VMEbus communication processor board (VCP) fills slot four. Slots five through eight are reserved for Ethernet or network



---

protocol boards (DDN/SNA) and host adapter boards. Note that the first host adapter board is installed in slot 8 (see Figure 2-4).

**[ WARNING ]** Board placement in a P/90 is very important. In slots 4 through 8, either a board or a jumper module must be installed. If there is an empty slot, a jumper must be inserted from the backside of the backplane.

**Table 2-6. P/90 Card Cage Slot Assignments**

Non-Ethernet		Ethernet	
Slot #	Board	Slot #	Board
8	Host Adapter 0	8	Host Adapter 0
7	Host Adapter 1	7	Host Adapter 1
6	Host Adapter 2	6	Host Adapter 2, Ethernet or DDN/SNA
5	Host Adapter 3	5	Ethernet or DDN/SNA
4	VCP	4	VCP
3	Arbiter	3	Arbiter
2	68020 CPU	2	68020 CPU
1	Memory	1	Memory

### **P/90 Expansion Cabinets**

To provide the P/90 with disk and tape expansion capabilities, the P/90 can accommodate up to two expansion cabinets. These expansion cabinets are expected to be available in the future.

### **Additional Information**

For additional information on the P/90s features, see the *P/90 Installation Guide*. The *P/90 Maintenance Manual* provides additional technical information, cabling requirements, and preventive maintenance procedures.

---

## **P/95 System**

This section describes:

- component parts of the P/95
- P/95 card cage slot assignments
- expansion capabilities of the P/95
- Arbiter board capabilities
- additional information about the P/95 computer system

### **P/95 System Description**

A minimum, single-cabinet P/95 system is composed of the following components:

- 68020 processor board
- Up to three memory boards
- VMEbus Communications Processor I/O Controller (VCP)
- Cartridge tape controller
- AMSP disk controller
- Power supply
- One 142Mb or 145Mb 8-inch hard disk drive
- 60Mb cartridge tape drive

### **P/95 Card Cage Slot Assignments**

Tables 2-5 through 2-7 illustrate card cage slot assignments for the P/95.

---

**[ WARNING ]** Board placement in a P/95 system is extremely important. The first five slots contain the CPU, memory, and arbiter boards. Either a board or a jumper module can be installed in slots 6 through 17. If there is an empty slot, a jumper must be inserted from the backside of the backplane.

Also, because of the size of the modem on the arbiter board, a board must not be installed in slot 6.

**Table 2-7.** P/95 Card Cage Layout — Stand-alone System

<b>Slot #</b>	<b>Board Name</b>
1	Memory Board 2
2	Memory Board 1
3	Memory Board 0
4	CPU
5	Arbiter
6	no board
7	VCP 0
8	VCP 1
9	VCP 2
10	VCP 3
11	DDN or SNA
12	ethernet
13	no board
14	no board
15	disk controller
16	no board
17	cart. tape controller

---

**Table 2-8.** P/95 Card Cage Slot Assignments — One Expansion Cabinet

<b>Slot #</b>	<b>Board Name</b>
1	Memory Board 2
2	Memory Board 1
3	Memory Board 0
4	CPU
5	Arbiter
6	no board
7	VCP 0
8	VCP 1
9	no board
10	Ethernet
11	disk controller
12	disk controller
13	disk controller
14	disk controller
15	no board
16	reel-to-reel tape controller
17	cart. tape controller

---

---

**Table 2-9. P/95 Card Cage Slot Assignments — Two Expansion Cabinets**

<b>Slot #</b>	<b>Board Name</b>
1	Memory Board 2
2	Memory Board 1
3	Memory Board 0
4	CPU
5	Arbiter
6	no board
7	VCP 0
8	VCP 1
9	VCP 2
10	VCP 3
11	SCSI, DDN, or SNA
12	disk controller or ethernet
13	disk controller/Multibus adapter
14	disk controller
15	disk controller
16	reel-to-reel tape controller
17	cart. tape controller

---

### **P/95 Expansion Cabinets**

Expansion Cabinets are available for the P/95 computer. The cabinets are mounted on casters.

The P/95 can accommodate up to four expansion cabinets. Each P/95 expansion cabinet bolts to the adjoining P/95 cabinet. Cables within each expansion cabinet connect to an adjoining cabinet through the use of connectors. These connectors are mounted on a panel in a duct connecting the cabinets.

Each expansion cabinet can contain the following:

- reel-to-reel streaming tape drive (no disk drives)
- reel-to-reel streaming tape drive plus
  - a. one 443Mb disk drive or

- 
- b. two 443Mb disk drives
  - One to four 443Mb fixed disk drives (no tape).

### **P/95 Arbiter Board**

The Arbiter board performs the following functions:

- Bus clock signal generation
- Priority coding/decoding
- Reset signal generation
- Power control for the P/75 and Expansion cabinets
- Power fail signal generation
- Special Interrupt notification
- Two levels of temperature sensing:  $45^{\circ}\text{C} \pm 5^{\circ}$ , and  $70^{\circ}\text{C} \pm 5^{\circ}$ .
- D.C. voltage level monitoring
- Uninterruptable Power Source Interrupt connection
- On board 300/1200 bps modem (North American systems only)
- LED board status indicators

Note that the built-in auto-dial / auto answer modem is a standard P/95 feature facilitating isolation of hardware and software problems (North American systems only).

For additional information on the P/95 Arbiter board, including LED indicator values, see the *P/95 Maintenance Manual*.

### **Additional Information**

For additional information on the P/95s features, see the *P/95 Installation Guide*. The *P/95 Maintenance Manual* provides additional technical information, cabling requirements, and preventive maintenance procedures.

---

## **Hardware Comprising Plexus Systems**

Each Plexus system is comprised of component parts such as:

- Backplane
- Power Supply
- Job Processor
- Memory Boards
- Communication Processors
- Disk and Tape Controllers
- Disk Drives
- Tape Drives
- Options

These component parts are described in the following sections.

---

## Backplane

Your Plexus system contains various boards including: CPU, memory board(s), communication processor(s), disk controller(s), and tape controller(s). Additional boards, such as network boards, are available from Plexus as options. These boards are connected to each other by another board set perpendicularly called a backplane. This backplane, containing the system I/O bus decides who gets to talk and when amongst the several boards. The bus provides a path for connecting the I/O boards and provides rapid timed transfer of data between boards. Table 2-10 describes the system I/O bus in Plexus systems.

**Table 2-10. System I/O Bus Summary**

System Name	System I/O Bus
P/35, P/55, P/60, P/75 P/90 and P/95	IEEE-796 Multibus VMEbus

### P/90 and P/95 VMEbus Backplane

The VMEbus relies on a daisy chain to pass a bus grant signal. The signal starts at the edge of the backplane and is passed from board to board until it arrives at the board requesting the VMEbus.

On the P/90, the VMEbus begins at slot 8 and goes through slot 4. A memory board, the 68020 job processor, and the Arbiter board reside in slots 1 through 3. On the P/95, the VMEbus begins at slot 17 and goes through slot 6. Memory boards, the 68020 job processor, and the Arbiter board reside in slots 1 through 5.

If any slot (6 through 17 on the P/95 or slot 4 through 8 on the P/90) is not occupied by a board, a jumper module must be inserted into the upper connector of the unused slot in the back of the backplane. This requirement ensures that the bus grant signal will be passed to all boards.



---

## Power Supply

A power supply unit provides the needed voltages for the boards via the backplane. Also, fans provide necessary ventilation.

The P/35, P/55, P/60, P/75, P/90 and P/95 are equipped with a multiple-output, switching power supply providing overvoltage and short-circuit protection. The P/75 power supply is controlled by the Common Circuits Board (CCB). The P/90 and P/95 power supply is controlled by the Arbiter board. See your Maintenance Manual for voltage and amperage information.

## Job Processor

The Plexus systems described in this manual are based on the Motorola 68000 family of job processors. The job processor, which provides processing power for your Plexus system, handles all communications between different I/O boards on the I/O processor bus.

These job processors are 32-bit (68020) or 16/32-bit (68000) job processors. The 68020 job processor in the P/90 and P/95 systems provides the highest CPU performance compared with other Plexus Multibus 68020 systems. Table 2-11 lists the standard configuration for the CPU board.

**Table 2-11. Job Processors in Plexus Systems**

<b>System</b>	<b>Standard Job Processor</b>
P/35	Motorola 68000
P/55	Motorola 68020 (12.5MHz clock)
P/60	Motorola 68020 (12.5MHz clock)
P/75	Motorola 68020 (12.5MHz clock)
P/90	Motorola 68020 (20MHz clock)*
P/95	Motorola 68020 (20MHz clock)*

\* At a later date the 68020 running at 25MHz is expected to be available

For additional information on systems architecture, see the *Technical Summary* manual. Contact your sales representative for a

---

copy of the *Technical Summary*.

## **Memory Boards**

This section discusses memory boards in Multibus and VMEbus-based systems. For additional information, see the *Technical Summary* manual.

### **Memory Boards in Multibus Systems**

The memory subsystems used in Multibus Plexus systems include: .5Mb (in older systems), 1Mb, 2Mb, and 4Mb boards. Each memory board provides random access memory (RAM) for your system. This memory is volatile memory. That is, when the system is shut down, what is in the system's memory is not remembered.

### **Memory Boards in VMEbus-Based Systems**

The P/90 system can contain up to one memory board. The P/95 system can contain up to three memory boards (48Mb maximum). Memory boards of differing capacities can be used in the same system.

Memory boards in VMEbus-based systems take advantage of the higher densities provided by SIMMs (Single Inline Memory Modules). Each SIMM contains 9 surface mounted DRAMs on a small PC board. With 256Kbit DRAM, a memory board can store 16Mb. Each SIMM is installed in socket. Presently, each memory board in a VMEbus system has a maximum capacity of 16Mb (using 256Kbit DRAM). The 16Mb memory board can be populated in the following capacities: 2Mb, 4Mb, 8Mb, and 16Mb.

### **Memory Board Overview**

Multibus and VMEbus memory boards are very similar. For simplicity of discussion, each board will be treated as identical. Each Plexus system has a minimum of one memory board. Each memory board contains quite a few individual memory chips adding up to the total amount of available random access memory (RAM), depending on the memory board installed in your system.

---

The memory is accessible by any of the other boards through direct memory access (DMA) and the I/O Processor Bus. DMA is a procedure designed to gain direct access to main memory and transfer data to and from main memory without involving the CPU. Besides interfacing directly through DMA, the memory board also interfaces via the CPU through the high speed memory bus. This bus is also contained on the backplane.

Plexus systems feature automatic single-bit error correction on all memory systems. This is done by the hardware and does not delay access to and from memory.

## Communication Processors

The number of users each system can support is directly related to the communications board(s) installed. The Intelligent Communication Processor (ICP), Advanced Communication Processor (ACP), and VMEbus Communication Processor (VCP) provide communication links between your Plexus system and serial devices (terminals, serial printers, modems) and parallel devices (parallel printers). Table 2-12 describes the standard communication processors available in Plexus systems.

**Table 2-12.** Communication Processors in Plexus Systems

<u>System Name</u>	<u>Communication Processors</u>
P/35	ICPs
P/55	ACPs
P/60	ICPs or ACPs
P/75	ACPs
P/90 and P/95	VCPs

Each ICP supports eight (8) serial ports and 1 parallel port. Each ACP supports sixteen (16) serial ports and one parallel port. The number of parallel and serial ports each VCP can support is described later in this section.

Although the P/60 system is shipped with ICPs as standard communications processors, you can upgrade ICPs with ACPs.

Installing one or more ACPs in a P/60 will also require that you install a cooling modification. Contact Technical Software Support at Plexus Computers for additional information. The following table lists the maximum number of ICPs, ACPs, and the maximum number of users for each system. Note that all systems can mix ICPs and ACPs in the same system.

**Table 2-13.** Number of Serial I/O Ports — Multibus Systems

<b>Plexus System</b>	<b>Maximum # of ICPs</b>	<b>Maximum # of Serial I/O Ports</b>	<b>Maximum # of ACPs</b>	<b>Maximum # of Serial I/O Ports</b>
P/35	2	16	2*	32
P/55	n/a	n/a	2	32
P/60	5	40	5*	80
P/75	n/a	n/a	5	80

n/a = not available; \* = upgradable option

As indicated in Table 2-6, with five ICPs the P/60 has 40 serial I/O ports.

On VMEbus systems, serial and parallel ports are provided by installing line interface cards (Link cards) and VMEbus Communications Processors (VCPs) in your P/90 or P/95. The same types of Link cards are available for the P/90 and the P/95. The P/90 can support up to 32 serial ports and 4 parallel ports.

The actual number of serial and parallel ports your P/90 or P/95 system can support depends upon the type of Link cards installed. The types of Link cards are offered include:

- DB-9 Link card (eight serial ports per card with DB-9 connectors).
- P-4 Link card (four parallel ports per card). The connectors on the P-4 Link card are IBM PC compatible.

For pinout information, see the chapter "Setting Up I/O Devices". For additional information, see your Installation Guide.

---

## **Intelligent Communications Processor**

The Intelligent Communications Processor (ICP) is a powerful 16-bit processor module designed to handle serial and parallel I/O tasks in P/35s and P/60s. The ICP provides the buffering and processing required to support high-speed communications with terminals, modems, printers, and other serial devices.

The ICP, a double-height Multibus module, is controlled by a 16-bit processor with 48Kb of memory. Each ICP controls eight (8) RS232C serial ports and one (1) Centronics<sup>†</sup>-type parallel printer port. Each serial port has full modem support and a maximum transfer rate of 19.2Kb. The serial ports are capable of supporting the asynchronous and bisynchronous protocols.

## **Advanced Communications Processor**

The ACP is a double-height Multibus card which manages communications for Plexus' P/55 and P/75 systems. The ACP is available for the P/95 system and is also available as an option for the P/60 system. The ACP, controlled by a Motorola 68000 processor and with 512Kb memory, allows segments of the operating system and customized communications and terminal handling programs to be downloaded to the ACP and executed locally.

Each ACP controls sixteen serial ports (terminals or modems) and one parallel (Centronics-type) port. The serial ports are RS232C compatible and have the modem control lines necessary to support standard asynchronous and synchronous modems. These ports are capable of asynchronous or synchronous protocols at software-selectable rates up to 19.2Kbaud. All sixteen serial ports can operate at the maximum rate simultaneously since each port has its own DMA channel.

---

† Centronics is a trademark of Centronics.

---

## VMEbus Communications Processor

The VMEbus Communications Processor (VCP) manages communications for the P/90 and P/95 systems. The P/90 system supports up to one VCP. The P/95 system supports up to four VCPs. The VCP provides an interface between the Plexus job processor. It also provides serial ports (terminals, serial printers, and modems) and parallel ports (IBM PC<sup>†</sup> compatible connections). The VCP consists of a main logic board and up to 5 optional I/O boards, Link cards). See the section "Serial and Parallel Ports" in this chapter for additional information on Link cards.

The VCP has two local processors which coordinate the movement of information between the job processor and attached devices. The Motorola 68020 main processor is aided by a high speed, bit slice processor performing the Direct Memory Access function.

The VCPs data bus is connected to the I/O cards (Link) forming the I/O of your system. Since the I/O is connected directly to the data bus, a scheme has been developed to translate a device name, such as `/dev/tty5`, into a physical port on a VCP.

### Additional Information

See the chapter "Board Descriptions" in the *P/95 Maintenance Manual* and the *P/90 Maintenance Manual* for information on Link card and VCP switch settings.

---

<sup>†</sup> IBM PC is a trademark of International Business Machines Corporation.

---

## Disk and Tape Controllers

Table 2-14 illustrates the disk and tape controllers available for Plexus systems.

**Table 2-14. Disk and Tape Controllers Available**

<b>System</b>	<b>Disk and Tape Controllers</b>	<b>Function</b>
P/35	IMSP	Controls disk & cart. tape
P/55	IMSP	Controls disk & cart. tape
P/60	IMSP EMSP CPC tape cont.	Controls disk drives (stand-alone) Controls disk drives (expan. cab.) Controls reel-to-reel tape
P/75	IMSP  EMSP CPC tape cont.	Controls disk & cart. tape (stand-alone) Controls disk drives (expan. cabs.) Controls reel-to-reel tape (reel-to-reel tape drive in expansion cabinets only)
P/90	disk cont. cart. tape cont. SCSI	Controls disk drives Controls cart. tape drive Controls disk & cart. tape controllers in main cabinet.
P/95	AMSP cart. tape cont. reel-to-reel tape cont.	Controls disk drives Controls cart. tape (stand-alone) Controls reel-to-reel tape (expansion cabinets only)

Each controller is described in the following subsections.

---

### **Intelligent Mass Storage Processor (IMSP)**

The IMSP is an intelligent disk and cartridge tape controller containing its own Z8001 job processor. It is used in Multibus systems. It receives commands from the CPU to move blocks of data between system memory (RAM) and the disk drives or cartridge tape drive. The processor's 256Kb address space is organized as follows: 16Kb local ROM, 128Kb local RAM, and 64Kb shared RAM. The 128Kb of local RAM is used to buffer a number of sectors, to decrease the number of disk accesses when the system experiences a heavy processing load. These buffers store the information from the disk and pass it to each process as if it were the only process using disk.

The IMSP uses an industry-standard SMD type disk interface and also controls the cartridge tape drive (QIC02 type cartridge tape interface). It records and reads using the QIC-11 tape format. The intelligent cartridge tape drive performs many of the functions normally required of a tape controller. It communicates with the IMSP over eight data lines and eight control lines.



---

## **Enhanced Mass Storage Processor (EMSP)**

The performance of larger Plexus Multibus systems (P/60 and P/75) is greatly improved through the use of additional disk drives in Expansion Cabinets. In the P/75, one to four EMSP boards are used to interface with hard disk drives located in the main or Expansion Cabinets.

The EMSP controller is a disk controller only. The EMSP utilizes an advanced job processor-based design to achieve high throughput and implement advanced performance features such as overlapped seek operations, overlapped data transfers, and 32-bit error correction codes (ECC) to ensure data integrity. Each EMSP can support a maximum of four disk drives. Multiple EMSPs can be run in parallel when very fast access or support for very large amounts of data storage is required.

## **AMSP Disk Controller**

The disk controller for the P/95 system is the Advanced Mass Storage Processor (AMSP), which is a standard size VMEbus board. The AMSP enables the P/95 to utilize high capacity disk drives employing the Enhanced Storage Module Device (ESMD) interface. The AMSP utilizes a 68000 job processor-based design to achieve high throughput and implement advanced performance features such as overlapped seek operations, overlapped data transfers, and a 32-bit ECC error control to ensure data integrity. Initially, each AMSP can support up to 2 disk drives. At a later date, each AMSP can support up to 4 disk drives.

Thus, the P/95 system can initially support 10 disk drives. At a later date, the P/95 system can support up to 16 disk drives. Multiple AMSPs can be run in parallel when very fast access or support for very large amounts of data storage is required.

Disk errors are handled by the built-in error correction facilities of the AMSP controller. During a disk read operation, the disk controller can detect and automatically correct an erroneous burst up to 11 bits in length. Error detection and correction is performed by a 32-bit error code appended to each sector ID or data field when the field is written to the disk.

---

### **SDC-Fixed ESDI Disk Controller — P/90**

The SDC-Fixed ESDI disk controller is a member of a family of disk controllers developed around the SCSI disk controller (SDC). The SDC-fixed ESDI disk controller is an intelligent high level interface between the SCSI bus and the ESDI disk drives enabling electronic controls of the SCSI bus, automatic handling of bus protocol, and retry/ECC correction for the fixed drives.

### **1/4-inch Cartridge Tape Controller — P/90**

The SCSI host adaptor controls the ¼" cartridge tape controller. The ¼" cartridge tape controller controls the 60Mb cartridge tape drive in the P/90 system.

### **1/2-inch Cartridge Tape Controller — P/90**

The SCSI host adaptor controls the 1/2-inch tape controller. The tape controller controls 1/2-inch cartridge tape drive which is expected to be available at a later date.

### **1/4-inch Cartridge Tape Controller — P/95**

The ¼" cartridge tape controller controls the 60Mb cartridge tape drive in the P/95 system using the QIC-02 interface. It records and reads the QIC-11 tape format.

### **Reel-to-Reel Tape Controller — P/60 and P/75**

On Multibus systems, the ½" reel-to-reel tape drive is controlled by the CPC controller. It has a Pertec-type interface. This controller has its own job processor and a local memory buffer.

### **Reel-to-Reel Tape Controller — P/95**

The ½" reel-to-reel tape controller controls the ½" reel-to-reel tape drive in the P/95 system. It has a Pertec-type interface. This controller has its own microprocessor and a local memory buffer.

---

## Disk Drives

Plexus systems use a variety of high technology Winchester disk drives starting with 5 ¼" drives and advancing to 8" and 9" drives (14" drives on older Plexus systems). For every system, there is a minimum of one disk drive.

P/90 drives use the Enhanced Small Device Interface (ESDI). P/35, P/55, P/60, and P/75 drives use the Storage Module Device (SMD) interface. The SMD controller is an industry standard interface which allows the addition of any SMD type disk drive. Up to four (4) disk drives can be daisy-chained together on each IMSP, EMSP, or AMSP controller. The P/95 drives use the Enhanced Storage Module Device (ESMD) interface.

## Optical Disk

The Plexus optical disk is a Write-Once Read-Many (WORM) optical disk designed to archive and retrieve very large amounts of data (1Gbyte per disk side). The stand-alone optical disk is supported on the P/55. The P/75 and P/95 systems can support up to 8 rack mountable optical disk drives. For installation and user instructions, see the *Optical Disk User's Manual*. At a later date, 5 ¼" optical disk is expected to be available for the P/90 system.

On Multibus systems, a SCSI adaptor is required to use optical disk. On the P/95, the Multibus adaptor and the SCSI adaptor are required to use optical disk.

---

## Tape Drives

Your Plexus system comes with at least one tape drive. Use tape drives for off-line storage (i.e. backups). The following table lists the default tape drive configuration for each Plexus system.

**Table 2-15.** Tape Drives in Plexus Systems

System	Tape Drive(s)
P/35	45Mb 1/4-inch cartridge tape drive is standard.
P/55	60Mb 1/4-inch cartridge tape drive is standard.
P/60	1/2-inch reel-to-reel tape drive is standard. A stand-alone 1/4-inch cartridge tape drive (optional) is also available (60Mb or 45Mb).
P/75	60Mb 1/4-inch cartridge tape and/or up to four 1/2-inch reel-to-reel tape drives in the Expansion Cabinets.
P/90	60Mb 1/4-inch cartridge tape drive is standard.
P/95	60Mb 1/4-inch cartridge tape and/or up to four 1/2-inch reel-to-reel tape drives in the Expansion Cabinets.

## Tape Drive Features

The tape drive features are described in this subsection.

### P/35

The standard tape drive for the P/35 is a 45Mb 1/4" cartridge tape drive. This drive uses high-density tape cartridges to store information. The cartridge tape drive features streaming operation.

### P/55

The standard tape drive for the P/55 is a 60Mb 1/4" cartridge tape drive. This drive uses high-density tape cartridges to store information. The cartridge tape drive features streaming operation.

---

### **P/60**

The standard tape drive on the P/60 is a ½" reel-to-reel 1600 bpi unit featuring horizontal mounting, automatic threading, and dual-mode operation. In streaming mode, which is used for disk backup, the tape drive stores 46Mb of data in 4.8 minutes on a 10 ½" 2400-foot reel of tape. In normal start/stop mode, the ANSI/IBM-compatible drive provides a convenient method for exchanging data with other computer systems.

### **P/75 and P/95**

The standard tape drive for the P/75 and the P/95 is a 60Mb ¼" cartridge tape drive. This drive uses high-density tape cartridges to store information. The cartridge tape drive, which features streaming operation, also provides media compatibility with smaller Plexus systems.

### **P/90**

The standard tape drive for the P/90 is a 60Mb ¼" cartridge tape drive. This drive features streaming operation for fast disk backup and provides media compatibility with other Plexus systems.

---

## Storing Data on Tape

Tapes are the best method of storing large data files, when that data is not in constant use. Also, tapes are used to backup your complete disk file system as insurance against disk disasters. These can include user mistakes in removing files, total corruption of an entire file system, hardware problems with the disk units, and other such critical situations. Two methods of storing data on tape are discussed in the following paragraphs.

The first way, known as archiving, is typically invoked using the **cpio** or **tar** command (**cpio** is recommended). This method should be used for temporarily storing large amounts of information that take up disk space when freeing up disk space is a necessity. It is also used to transfer files of any type (except device files) from one Plexus system to another.

The second means for using tape is for backup dumps to guard against losing important user and system files in the case of a disk or other system failure and is invoked with the Sys3 command, **/etc/dump**. Note that there are two **dump** commands in the Plexus release of Sys5. Plexus carried over the Sys3 **dump** program. It resides in the **/etc** directory. Use **/etc/dump** to dump file systems to tape. The new Sys5 command **/bin/dump** is an octal dump. Backing up writes an image of what is on the disk onto the tape so that, if necessary, an entire disk file system can be reconstructed easily.

Tape archiving, on the other hand, writes out files and directories as they appear to the user. As we will see in the chapter describing file systems, the organization of files and directories that the user sees is much different than the way things are organized on the disk itself. Both tape archiving using **cpio** and backups to tape using the Sys3 command **/etc/dump** will be described later in the chapter "Backup Procedures". Other commands which can be used in place of **/etc/dump** for backup purposes are **fbackup** (not available on VMEbus-based systems), **volcopy**, **finc**, and **dd**.

---

## Peripheral Devices

Peripheral devices are devices external to your Plexus system and include: terminals, modems, parallel printers, and serial printers. Following is a discussion of each peripheral device.

### Terminals

You will have one or more types of terminals connected to your system. These terminals will be the primary point of contact between the user(s) and the system itself. These terminals might be a CRT type terminal (TV screen type), or they might be a hard-copy device (typewriter-style that prints the output on paper).

Each of your particular type of terminal will probably have an on-line/off-line, on-line/local or conv mode switch which must be in the on-line position to communicate with the computer. Different types of terminals operate at different speeds (baud rates). These speeds are usually controlled by a switch that is located on the terminal or is programmed into the terminal.

The asynchronous ASCII terminals are connected by an RS232 cable to any one of the serial ports. How to connect terminals is described in the chapter *Setting Up I/O Devices*.

Normal video display terminals usually operate at 9600 baud, full duplex, no parity, and 1 stop bit. Serial port (software) and terminal (hardware) characteristics can be set independently.

### Modems

Modems, either hard-wired or dial-up, can also be connected to the serial ports by means of a NULL modem cable. The physical port on the ICP itself, must also be altered to expect a modem connection (DCE) as opposed to a terminal (DTE). All ICP ports are configured for DTE connection at the factory. ACPs and VCPs are totally software configurable. How to configure your system for modem use (with ACPs and VCPs) is explained in the chapter *Setting Up I/O Devices*.

---

## Line Printers

Plexus systems support two kinds of printers: parallel and serial. These printer types will be discussed in the subsections that follow.

### Parallel Printers

A parallel printer is usually a high speed printer and is traditionally used to print out large amount of data for drafts or any other non-letter-quality purpose.

On Multibus systems, each ICP or ACP has one parallel port specifically designed to support parallel interface. This port is composed of a DB37-pin female connector. Pinouts for the cable will be described later in this manual. Parallel cables for Multibus systems can be ordered from Plexus. The printer itself can be of any manufacturer, but it must support a Centronics interface in order to function properly. All of the output settings for the printer are controlled purely by hardware switch settings on the printer itself. The Centronics interface standard was adopted to provide a common hardware interface to the parallel port driver.

On VMEbus-based systems, the parallel printer cable needed is a 25-pin IBM PC compatible cable. This IBM PC compatible cable is available at any major computer store. The printer itself can be of any manufacturer, but it must support the IBM PC compatible interface in order to function properly. Parallel Link cards (P-4) are available from Plexus.

### Serial Printers

A serial printer is usually a lower speed, high print quality device used for correspondence and generating forms. You can connect a serial printer to any one of the system's serial (tty) ports just as you would connect a terminal. How to set up a serial printer is described in the chapter *Setting Up I/O Devices*.





## SOFTWARE COMPONENTS

---

This chapter:

- presents an overview of the Sys5 operating system.
- describes the three major file types.
- describes the Sys5 root file system structure, including a description of each directory under root ( / ).
- provides information on the block and character device numbers used in **/dev**.
- gives a brief explanation describing when to run stand-alone programs (**/stand** or from the release tape).

---

## Sys5 Operating System Overview

The Sys5 operating system is a package of software that manages the resources of your Plexus system. The software of any computer system is a collection of files. These files contain executable code which can be read and interpreted by the system to perform a set of functions.

The Sys5 operating system is divided into four major parts:

- the kernel
- the file system
- Sys5 utilities
- the shell

Following is a brief description of each part of the operating system.

The kernel	includes the scheduler and memory management functions, and is the basic resident software on which the entire system relies. It is the only part of the system permanently resident in memory. The job of the kernel is to control user processes and manage system resources. The Sys5 kernel, <b>unix</b> , is a program that is loaded when a Plexus system is started. It runs continuously until shutdown.
The file system	is the organization scheme for storing data on disk. The Sys5 file structure is organized in a tree-like structure, comprised of files and directories.
The Sys5 utilities	consist of executable commands and libraries that enable the Sys5 user to perform a variety of system and user tasks.

---

The shell

is the command interpreter. The shell listens to your terminal and translates users' requests into actions on the part of the kernel and the many utility programs. After the shell finds programs that the user wants to execute, the shell passes those tasks to other parts of the operating system for scheduling, input, output, and execution. Thus, the shell is the primary user interface to utilities and applications.

Sys5 is an interactive operating system. As you enter commands, the system obeys the commands and displays appropriate responses. Sys5 is a multi-tasking operating system. That is, the operating system can schedule and perform many tasks simultaneously. Sys5 is also a multi-user operating system: more than one person can use the system simultaneously.

**[ NOTE ]**

The kernel should only be modified by a Sys5 guru (someone who knows what (s)he is doing). The kernel will only need to be changed if you install new hardware, if a new version of the operating system is installed, or if system user patterns change and you discover that you need new or additional system resources. It is strongly recommended that you contact Plexus or your support staff for specific instructions on kernel modifications before attempting it yourself no matter how you feel about your expertise!

---

## File Types

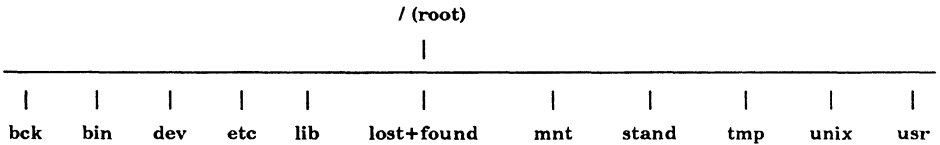
As mentioned previously, the Sys5 file system is the organizing structure for data. The file system contains three types of files:

- ordinary files      are files you create for productive system work. Simply put, a file is a container for information. Ordinary files can contain executable object code, shell scripts, binary data, or ASCII text. You can create ordinary files by using one of the utilities (such as an editor) or by redirecting I/O.
- directories          are files which group other files and directories. You can think of a directory as a container for other directories and files. Directories are created with the **mkdir** command. A directory contained within another directory is called a subdirectory. Subdirectories can contain files and more subdirectories. In Sys5, directories supply the matrix which superimposes a hierarchical structure on the file system.
- special files      are device names. Special files are created using the **mknod** command and are listed in the **/dev** directory. Some examples of special files (device names) are: **/dev/tty4** (serial device tty4), **/dev/rrm/0m** (reel-to-reel tape drive 0, raw device), **/dev/rpt/0mn** (cartridge tape device 0, raw device with no rewind), **/dev/pp0** (parallel port 0).

---

## Sys5 Root File System Structure

This section introduces you to the Sys5 main directory structure. It gives you a general idea of how the system hierarchy of directories, ordinary files, and special files are organized. The Sys5 file system is organized into directories structured in a hierarchical scheme, converging at the top (inverted tree) in the directory called root ( / ). Figure 3-1 illustrates the root file system structure.



**Figure 3-1.** Root File System

Figure 3-2 illustrates the result of listing the contents of the root ( / ) directory with the long ( -l ) option ( ls -l / ). Your display might be different.

```

drwxr-xr-x   2   root  sys   32           Aug 16 11:02   bck
drwxrwxr-x   2   bin   bin 1504          Mar  5 14:46   bin
drwxr-xr-x   9   root  sys 2064          May  8 11:43   dev
drwxrwxr-x   3   root  sys 1600          May 27 10:53   etc
drwxrwxr-x   3   bin   bin 336           Dec 20 09:03   lib
drwxrwxrwx   2   bin   bin 1024          Apr 25 16:26   lost+found
drwxr-xr-x   2   root  sys 32           Aug 16 07:14   mnt
drwxrwxr-x   2   bin   bin 240           Dec 20 09:05   stand
drwxrwxrwx   2   root  sys 816           May 27 11:26   tmp
-rwxrwxr-x   1   root  sys 294659        Dec 20 12:41   unix
drwxrwxr-x  20   bin   bin 368           Feb 20 17:17   usr

```

**Figure 3-2.** Listing the Contents of the Root Directory

The line items beginning with **d** are directories; line items beginning with a dash (-) denote files. In the / (root) directory, usually the only file is **unix**, which is the operating system program, or kernel. An explanation of the previous listing is illustrated in Figure 3-3.

Protections	#	Indiv.	Group	#	Date	File
	links	owner	owner	bytes	Created or Modified	name
drwxrwxr-x	3	root	sys	1600	May 27 10:53	etc
-rwxrwxr-x	1	root	sys	294659	Dec 20 12:41	unix

**Figure 3-3.** Listing the Root File System With the Long Option

## Root File System Description

The major root subdirectories (i.e. **bin**, **dev**, **etc**, **lib**, **stand**, **usr**) are explained in detail below because you will use them frequently. These follow a list of brief descriptions of the remaining items in the root directory.

- bck**                    System backup mountstub directory, otherwise empty.
- mnt**                    This is an empty directory (i.e. no subdirectories or files are under it). This is a convenient directory into which mount stub directories can be placed.
- lost+found**            This is a directory that the file system check program, **fsck**, uses to deposit files that somehow are disconnected from their original directory. It provides a place where the **fsck** program can place the orphaned file. If no orphan files are present in **lost+found** with numerical names, it is usually empty. **THERE SHOULD BE ONE lost+found** directory for EACH **sys5** file system created.

---

**unix** The file **unix** is the image of the Sys5 operating system. It is read into main memory when the operating system is loaded into memory (i.e. when you boot the system). This is an object file and should not be displayed or modified. If it is removed, you will not be able to reboot the system after a failure, power-up, or system reset. You will have to boot from the release tape.

**stand** This directory contains executable programs used to examine, check, and repair the system when it cannot be booted in the normal manner. These stand-alone programs can only be executed from the boot prompt ( : ) and never when the system has been successfully brought up.

**tmp** This directory is the system scratchpad. Certain applications (i.e. text editors, spreadsheets, database management systems, and compilers) require this writable directory to create temporary working files to store data during execution. Applications usually remove these tmp files after execution. This directory will most likely be empty if no users are logged in to the system.

The files in **/tmp** are assigned a unique file name by the application (usually a prefix followed by a unique number). File names that are prefixed with an "Ex" and "Rx" are temporary files created with one of Sys5's editors. By looking at the ownership of the file, the system administrator can tell which user is currently using the file.

Take care to not remove these files while the unsuspecting user is executing a program which uses them. Because files in **/tmp** are cleaned out during system startup (i.e. the contents of **/tmp** are purged when the system is re-initialized), DO NOT PUT ANY IMPORTANT FILES IN THIS DIRECTORY.



---

## **/bin Directory**

This directory contains executable files only, which comprise about half of the Sys5 utilities. Because these files are compiled programs, you cannot display or edit them. These programs are executed at the shell command prompt. The commands which reside in this directory represent the most commonly used commands for Sys5 system users. Some of the commands in the bin directory include **cat**, **ls**, and **date**. As the system administrator, you can add programs to the **/usr/bin** directory or you can create your own bin directory in **/usr** (i.e. **/usr/local** or **/usr/local/mybin**). Because new releases overwrite **/bin**, keep your own programs in **/usr/local** or **/usr/local/mybin**.

## **/dev Directory**

The **/dev** directory contains all of your device files. Device files are referred to as special files. Sys5 references all of your system devices and peripherals through these special files. Some of the special files include: **/dev/kmem** (kernel memory), **/dev/ttyX** (serial device ttyX where X is the serial port number), **/dev/rpt/0m** (cartridge tape drive 0, raw device), **/dev/rrm/0mn** (reel-to-reel tape drive 0, raw device with no rewind), and **/dev/pp0** (parallel printer 0).

Those device entries with protection bits beginning with a **c** refer to raw devices (sometimes referred to as character devices). Raw devices handle an arbitrary number of characters at a time without processing them (i.e. not putting them in a buffer first). Device entries with protection bits beginning with a **b** denote block devices. I/O is conducted in blocks of characters (i.e. 1024 characters or bytes to a block). Device entries with protection bits beginning with a **d** refer to directories. Invoke **ls** of the directory to list the block or character subdirectory information.

---

Following is a sample listing of the dev directory. Depending upon the devices you have set up, your listing might look quite different.

```
crw-rw-rw-  1  root  other  30,  1  May 18 22:55  ac1
crw-rw-rw-  1  root  other  30,  2  May 18 22:55  ac2
crw--w--w-  2  root  other   0,  0  May 29 09:38  console
drwxrwxr-x  2  root  sys      336  Feb 16 12:00  dsk
```

**Figure 3-4.** Partial Listing of the /dev Directory

### Major Device Numbers

The major block and character device numbers are displayed in the fifth column of the `ls -l` listing and are defined in the file: `/usr/src/uts/m68/cf/conf.c`. Because there are two types of devices (block and character), there are two tables. The first table, `bdevsw` is located about half way through the `conf.c` file. This is a listing of the major block device numbers. The second table, `cdevsw`, which follows the `bdevsw` table, is a listing of the major character device numbers. To display the contents of `conf.c` on the screen enter:

```
more /usr/src/uts/m68/cf/conf.c
```

**[ NOTE ]** BE CAREFUL! The `conf.c` file is the main configuration file for the kernel. It defines some system table sizes and some parameters such as what device drivers are used in the system. Do not change any of the information in `conf.c` unless you have a good working knowledge of what you are doing! Use `cat` or `more` to view the file's contents.

---

### Major Block Device Numbers (Multibus)

Following is a description of the major block device numbers for Multibus systems. Before creating new devices in the `/dev` directory, check this list with the file `conf.c`.

- 0        The **mi** driver (`/dev/dsk/XsXX`). This is a generic disk driver for mirrored and non-mirrored disks.
- 1        The **mt** driver, (`/dev/mtX`). This is not presently used.
- 2        The **pd** driver, (`/dev/dsk/XsXX`). This controls the IMSP disk drives.
- 3        **RAM** driver, (`/dev/ram/X`). This driver allows you to use system memory as a disk.
- 4        Not currently used.
- 5        Not used.
- 6        The **xy** driver (`/dev/dsk/XsXX`) on Multibus systems. This controls the EMSP disk drives.  
For the P/95 system, this is the **sm** driver (`/dev/dsk/XxXX`), which controls the AMSP disk drives.
- 7        Not currently used.
- 8        Not currently used.
- 9        The **od** (optical disk) driver (`/dev/od/XsXX`). This block device controls the optical disk drive.
- 10 — 14   Reserved for user use.

---

### Major Character Device Numbers (Multibus)

Following is a list of current major character device numbers and a brief description of each device driver. Before creating new devices in the **/dev** directory, check this list with the file **conf.c**.

- 0        The **us** driver, (**/dev/console**). This drives the system console.
- 1        Not currently used.
- 2        The **mm** driver. This is the memory device. The minor device numbers control which chunk of memory to use, kernel data space, data sink, etc. Special file names include: **/dev/mem**, **/dev/kmem**, **/dev/mbiomem**, **/dev/mbmem**, and **/dev/liomem**.
- 3        The **mi** driver. This is the generic disk driver for mirrored and non-mirrored disks. Special file names include: **/dev/rdkX** and **/dev/dsk/Xsxx**. *X* is the physical disk number (0, 1, etc), *xx* is the logical disk number (0-15).
- 4        The **mt** driver, (**/dev/rmtX**). This was supposed to be the generic tape driver for Sys5. However, it is at present not used. Use **/dev/rrm** or **/dev/rpt** instead.
- 5        The **pd** driver, (**/dev/rdsk/XsXX**). This controls disk drives connected to the IMSP.
- 6        The **pt** driver. This controls the cartridge tape drive. Special file names include: **/dev/rpt/Xm** (with rewind), **/dev/rpt/Xmn** (no rewind). *X* is the drive number.
- 7        **RAM** disk driver, (**/dev/rram/X**). This driver allows you to use system memory as a disk.
- 8        The **rm** driver. This controls the reel-to-reel tape drives. Special files include: **/dev/rrm/Xm** (with rewind), **/dev/rrm/Xmn** (with no rewind). *X* is the drive number.
- 9        The **xy** driver, (**/dev/rdsk/XsXX**). This controls the EMSP disk drives.

---

10 — 12 Reserved.

13 The **sy** driver for (**/dev/ttyX**) (*X* is the terminal number). This device represents the controlling terminal for any process.

14 The **ic** driver. The name that references this driver is (**/dev/icX**) (where *X* is the ICP number [0, 1, 2, etc]). This controls the ICP.

15 The **sp** driver, (**/dev/ttyX**). This is the serial port driver which controls the ICP terminal ports.

16 The **pp** driver, (**/dev/ppX**) (where *X* is the parallel printer port number). This is the parallel-port driver, which controls the ICP printer ports.

17 The **ncf** driver (no corresponding device). This is not available to you.

18 The **vpm** driver, (**/dev/vpmX**) (where *X* is the ICP number, i.e 0, 1, 2, etc.) This is used to communicate between a user program and an ICP executing virtual programmable memory.

19 The **tr** driver, (**/dev/trace**). This is used by some device drivers to communicate with user programs.

20 The **err** driver, (**/dev/error**). This device is used to pass errors from the kernel to a user process.

21 The **prf** driver, (**/dev/prf**). This device is an interface used to profile the operating system.

22 The **v tty** driver, (**/dev/vttyX**) (where *X* is the v tty number). This device is used to control the virtual terminal devices that are available in NOS.

23 The **im** driver, (**/dev/imX**) (where *X* is the IMSP number). This device controls the IMSP itself, rather than the peripherals attached to it.

24 The **hdlc** driver, (**/dev/hdlc**). This device controls the ICP for the hdlc protocol.

- 
- 25 The **ft** driver. This device controls caching cartridge tape. Some special file names are: **/dev/rft/0m** (rewind), **/dev/rft/0mn** (no rewind). This requires special disk setup for use.
- 26 Reserved.
- 27 Reserved.
- 28 The **crm** driver, (**/dev/crm/Xm**) (**rewind**) (**/dev/crm/Xmn**) (**no rewind**) where *X* is the drive number. This device controls a reel-to-reel tape in caching mode. It is recommended not to use this driver with the reel-to-reel cache drive, such as the GCR.
- 29 The **ccb** driver, (**/dev/ccb**). This device is for the P/75 only. It controls the common circuits board (CCB).
- 30 The **acp** driver, (**/dev/acX**) (where *X* is the ACP number (1, 2, 3, etc)). This device controls the ACP.
- 31 The **ACP** serial port driver, (**/dev/ttyX**) (where *X* is the ACP serial port number). This is the serial port driver which controls the ACP ports.
- 32 The **ACP** parallel port driver, (**/dev/ppX**) (where *X* is the port number 0, 1, 2, etc.). This is the parallel port driver which controls the ACP parallel printer ports.
- 33 Not supported.
- 34 Not supported.
- 35 Not supported.
- 36 The **ACP** dynamic download driver. There is no device name associated with this driver.
- 37 The **od** (i.e. optical disk) driver (**/dev/rod/XsXX**). This raw device controls the optical disk drive.
- 38 — 49 Reserved.
- 50 — 59 Reserved for user use.

---

### Major Block Device Numbers (VMEbus)

Following is a description of the major block device numbers for VMEbus-based systems. Before creating new devices in the `/dev` directory, check this list with the file `conf.c`.

- 0 The **mi** driver (`/dev/dsk/XsXX`). This is a generic disk driver for mirrored and non-mirrored disks.
- 1 Not used.
- 2 Not used.
- 3 **RAM** driver, (`dev/ram/X`). This device allows you to use system memory as a disk.
- 4 Not currently used.
- 5 Not used.
- 6 For the P/95, the **sm** driver (`/dev/dsk/XsXX`). This controls the AMSP disk drives.  
For the P/90, the **sd** driver (`/dev/dsk/XsXX`), which controls the MMSP disk drives.
- 7 Not currently used.
- 8 Not currently used.
- 9 The **od** (optical disk) driver (`/dev/od/XsXX`). This block device controls the optical disk drive.
- 10 Reserved.
- 11 Reserved.
- 12 Reserved.
- 13 Reserved.
- 14 Reserved.

---

## Major Character Device Numbers (VMEbus)

Following is a list of current major character device numbers and a brief description of each device driver for VMEbus-based systems. Before creating new devices in the `/dev` directory, check this list with the file `conf.c`.

- 0        The **us** driver, (`/dev/console`). This drives the system console.
- 1        Not currently used.
- 2        The **mm** driver. This is the memory device. The minor device numbers control which chunk of memory to use, kernel data space, data sink, etc. Special file names include: `/dev/mem`, `/dev/kmem`, `/dev/mbiomem`, `/dev/mbmem`, and `/dev/liomem`.
- 3        The **mi** driver. This is the generic disk driver for mirrored and non-mirrored disks. Special file names include: `/dev/rdkX` and `/dev/dsk/Xsxx`. *X* is the physical disk number (0, 1, etc), *xx* is the logical disk number (0-15).
- 4        The **mt** driver, (`/dev/rmtX`). This was supposed to be the generic tape driver for Sys5. However, it is at present not used. Use `/dev/rrm` or `/dev/rpt` instead.
- 5        Not used.
- 6        For the P/95, the **qc** driver. This controls the cartridge tape drive. Special file names include: `/dev/rpt/0m` (with rewind), `/dev/rpt/0mn` (no rewind). Plexus supports cartridge tape drive 0 only.  
  
For the P/90, the **st** driver. This controls the cartridge tape drive. Special file names include: `/dev/rmt/0m` - rewind, `/dev/rmt/0mn` - no rewind. Plexus supports cartridge tape drive 0 only.
- 7        **RAM** disk driver, (`/dev/rram/X`). This driver allows you to use system memory as a disk.



- 
- 8 The **tth** driver. This controls the reel-to-reel tape drives. Special files include: **/dev/rrm/Xm (with rewind)**, **/dev/rrm/Xmn (with no rewind)**. *X* is the drive number.
- 9 For the P/95, the **sm** driver, (**/dev/rdisk/XsXX**). This controls the AMSP disk drives.  
For the P/90, the **sd** driver (**/dev/rdisk/XsXX**), which controls the MMSP disk drives.
- 10 Reserved.
- 11 Reserved.
- 12 Reserved.
- 13 The **sy** driver for (**/dev/ttyX**) (*X* is the terminal number). This device represents the controlling terminal for any process.
- 14 Not used.
- 15 Not used.
- 16 Not used.
- 17 Not used.
- 18 Not used.
- 19 The **tr** driver, (**/dev/trace**). This is used by some device drivers to communicate with user programs.
- 20 The **err** driver, (**/dev/error**). This device is used to pass errors from the kernel to a user process.
- 21 The **prf** driver, (**/dev/prf**). This device is an interface used to profile the operating system.
- 22 Not used.
- 23 Not used.
- 24 Not used.
- 25 Not used.

- 
- 26 Reserved.
- 27 Reserved.
- 28 The **tth** driver, (**/dev/crm/Xmn** - no rewind, or **/dev/crm/X** - rewind) where *X* is the drive number. This device controls a reel-to-reel tape in caching mode. It is supplied for compatibility with other systems.
- 29 The **ccb** driver (**/dev/ccb**). This device controls the Arbiter board.
- 30 The **vcp** driver, (**/dev/vcX**) (where *X* is the VCP number (0, 1, 2, 3, etc)). This device controls the VCP.
- 31 The **VCP** serial port driver, (**/dev/ttyX**) (where *X* is the VCP serial port number). This is the serial port driver which controls the VCP ports.
- 32 The **VCP** parallel port driver, (**/dev/ppX**) (where *X* is the port number 0, 1, 2, etc.). This is the parallel port driver which controls the VCP parallel printer ports.
- 33 Not supported.
- 34 Not supported.
- 35 Not supported.
- 36 The **VCP** dynamic download driver. There is no device name associated with this driver.
- 37 The **od** (i.e. optical disk) driver (**/dev/rod/XsXX**). This raw device controls the optical disk drive.
- 38 — 47 Reserved.
- 48 **VCP** virtual system device table driver.
- 49 Reserved.

---

## Inside conf.c

This section discusses some constants in the file **conf.c**, the main configuration file for the kernel. A complete list of parameters for shared memory are defined. In addition, the constants NBUF, CDLIMIT, and NFILE are described.

**[ WARNING ]** Do not change any of the constants in **/usr/src/uts/m68/cf/conf.c** unless you have a good working knowledge of what you are doing! Contact Plexus Software Support for additional assistance.

- CDLIMIT** Defines the default maximum file size. It should yield a long integer, such as  $1L \ll 11$ , which defines how many kilobytes the file can be. It is currently 2048Kb, allowing files to grow to 2Mb. This can be changed to suit your environment. You can also use the **unlimit** program which allows you to create large files.
- MAXUP** Defines how many concurrent processes may be owned by a user-id (not superuser). If you have a common login for many users in an environment like data entry, you may need to increase this to avoid a situation where a program may abort and display the message "cannot fork", because the current user-id has MAXUP processes already in the process table. It should not be larger than NPROC. The current setting allows any user-id to have 25 processes active.
- NBUF** Specifies how many system buffers to allocate. If you increase the number, disk read/write time decreases, but amount of memory available to the user drops, so probably more swapping needs to be done. However, if you only have a few users, a lot of memory, and disk speed is important, you can

- 
- increase NBUF. This number must be a multiple of 4. Each entry in this table uses 26 bytes of memory, plus actual buffer space. NBUF 0 defaults to 10 percent of physical memory.
- NCALL** Determines the size of the system table that contains a list of functions that are to be invoked at some time in the future (in 50ths of a second increments) by the clock handler. The call-out table is used by terminal handlers to provide terminal delays and by various other I/O handlers. If this table overflows the panic message "timeout table overflow" will display, and your system will stop. Each entry in this table uses 6 bytes of memory.
- NCLIST** Specifies how many character list buffers to allocate. The buffers are linked together to form input/output queues for various character devices. The ICP kernel also contains some clist structures, which means this number is already about as small as it can be. Each entry in this table uses 28 bytes of memory.
- NFILE** Defines how large the file table is. Each entry in the table represents an opened file, either via open, creat, or pipe. When this table is full, the diagnostic *no file* displays, and no more files can be opened until some are closed. Each entry in this table uses 88 bytes of memory.
- NFLOCKS** Defines how many file locks can be active at any time. Remember that some files may have more than one region locked at a time, while others are never locked. This is a Plexus extension to standard *System V*, which did not support locking of files. Each entry in this table uses 10 bytes of memory.

---

NMOUNT	Defines how many mounted file systems there may be at one time. The root file system (/) is always the first entry in this table. When this table is full, no more mounts can be done, and an error will be returned. Each entry in this table uses 8 bytes of memory.
NPROC	Specifies how many process table entries to allocate. Each entry represents an active process. The swapper scheduler is always the first entry, and init is always the second. This number depends on the number of terminal lines and how many processes get created by each user. Each entry in this table uses 30 bytes of memory.
NTEXT	Specifies how many text table entries to allocate. Each entry represents an active read-only text segment, which was created using the <code>-i</code> or <code>-n</code> option to the C compiler ( <code>cc</code> ) or link-loader ( <code>ld</code> ). When the table overflows, the diagnostic message "out of text" displays. Each entry in this table uses 12 bytes of memory.
SHMALL	Specifies the maximum amount of shared memory that can be allocated system wide. The default value is 512 clicks, 250Kb.
SHMBRK	Specifies the number of clicks between the end of the data segment and the beginning of the first shared memory segment if the default starting address is used allowing the user to continue to use <code>sbrk(2)</code> or <code>brk(2)</code> . The default value is 16 clicks, 8 Kb.
SHMEM	Specifies whether to include shared memory code. A value of 0 indicates no shared memory. A value of 1 includes shared memory.
SHMMAX	Specifies the maximum size of a shared memory segment.

---

SHMMIN	Specifies the minimum size of a shared memory segment.
SHMMNI	Specifies the maximum number of shared memory segments in the system.
SHMSEG	Specifies the maximum number of shared memory segments in the system.
SMAPSIZ	Specifies how many entries there can be in the free-swap-blocks list. Each entry in this table represents a block on the swap device that is not in use. If the list overflows because this constant is too small, your system will crash in a very unpredictable manner. We recommend that you do not modify this constant from its current value. Each entry in this table uses 4 bytes of memory.
SYNCRATE	Allows some control over the sync call made by <code>/etc/init</code> every 30 seconds. If SYNCRATE is left at zero as shipped, there will be no change from previous releases. However, setting SYNCRATE to a positive value sets a limit to the number of buffers written when <code>/etc/init</code> does its periodic sync. This does not affect the normal operation of the sync command or the sync system call. Setting SYNCRATE reduces the number of unnecessary disk writes of blocks in temporary files, and improves the effects of the buffer aging algorithms. The suggested value for the SYNCRATE is 30 per disk drive.

---

## /etc Directory

The `/etc` directory contains most of the Sys5 commands and files that the system administrator uses in the setup and daily operation of the system. It can also be thought of as the superusers' toolkit, and it contains many editable text files which are used by the superuser to tailor the system. As such, programs in this directory are not accessible to regular users for execution. Although the `/etc` directory contains many system files and programs, the files introduced below are the major files that should concern you:

<b>bcheckrc</b>	<b>brc</b>	<b>checklist</b>	<b>cron</b>
<b>cshprofile</b>	<b>dconfig</b>	<b>ddate</b>	<b>dump</b>
<b>eccdaemon</b>	<b>fsck</b>	<b>fsdb</b>	<b>getty</b>
<b>gettydefs</b>	<b>group</b>	<b>grpck</b>	<b>init</b>
<b>inittab</b>	<b>labelit</b>	<b>mkfs</b>	<b>mknod</b>
<b>motd</b>	<b>mount</b>	<b>mmdir</b>	<b>ncheck</b>
<b>openup</b>	<b>passwd</b>	<b>profile</b>	<b>ps_data</b>
<b>pwck</b>	<b>rc</b>	<b>restor</b>	<b>setmnt</b>
<b>shutdown</b>	<b>termcap</b>	<b>ttytype</b>	<b>umount</b>
<b>utmp</b>	<b>wall</b>		

Following is a brief description of each file:

- bcheckrc**      The shell script **bcheckrc** performs all the necessary consistency checks to prepare the system to change into multi-user mode. It prompts to set the system date and to check the file systems with **fsck**.
- brc**            The shell script **brc** clears the mounted file system table, `/etc/mnttab`, and puts the entry for the root file system into the mount table.
- checklist**     Text file that tells **fsck** which file systems to check when no parameters are specified.
- cron**          The clock daemon program used to invoke programs at specified times.

- 
- C**
- csprofile** Sample login shell text used with the C shell (csh).
- dconfig** Program to configure logical disk partitions and parameters.
- ddate** Data file (used by the **/etc/dump** command) containing the date of the last full dump and information on incremental dump dates.
- dump** A Sys3 program used to backup entire file systems to tape.
- eccdaemon** Memory error logging daemon program.
- fsck** File system checking and repairing program. This program should be used each time the system is brought up and before using **/etc/dump** or one of the Sys5 backup utilities.
- fsdb** Interactive and thorough file system debugger program.
- getty** (get tty) This program runs as a process on each tty which is not in use. **getty** calls the login program which allows you to log in.
- gettydefs** This text file contains information used by **getty(1M)** to set up the speed and terminal settings for a line. It supplies information on what the login prompt should look like. Also, it supplies the speed to try if you indicate that the current speed is not correct by typing a <break> character.
- group** The text file containing information on group identification and members for the establishment of user groups.
- grpck** A program which checks for errors in the **/etc/group** file.
- init** The program to change system states (i.e. to go from single-user to multi-user).



---

<b>inittab</b>	A text file used by <b>init</b> to set tty speed and login characteristics for serial ports. <b>inittab</b> also starts certain initialization files such as <b>bcheckrc</b> , <b>brc</b> , and <b>rc</b> .
<b>labelit</b>	A program used to label file systems.
<b>mkfs</b>	A program used to make/remake a new (empty) file system.
<b>mknod</b>	A program used to make a device special file in the <b>/dev</b> directory.
<b>motd</b>	Message of the Day. A text file which is displayed to all users immediately after they log in.
<b>mount</b>	With arguments, <b>mount</b> attaches (mounts) disk file systems to Sys5 directories. Without arguments, it displays which file systems are currently mounted.
<b>mvdir</b>	A program used to move entire directories from one location to another within a file system.
<b>ncheck</b>	A program used to get a file's name from its inode number.
<b>openup</b>	A program daemon used to keep commonly accessed files opened continuously in memory. Because the inode is kept in memory when a file is open, it is faster to get to the file. Because the block addresses are in memory, the system can go directly to the data on the disk.

- 
- passwd** An important text file containing information which identifies to the system all users' accounts on your system. For backup purposes, it should be copied to **passwd.bak**. Your Plexus system cannot boot if **passwd** is corrupted or the file does not exist.
- profile** A shell command file which is executed when each user using **/bin/sh** logs in before their individual **.profiles** execute. This is a useful shell command file for setting characteristics which apply to all users running the Bourne shell (**/bin/sh**).
- ps\_data** Data which is created by **ps** as a file to locate tables in the kernel. If **ps** has problems it should be deleted. It will be recreated at the next use of **ps**.
- pwck** A program which reports possible errors in the **passwd** file.
- rc** A shell text executed by **init** when changing system states. It is responsible for starting all the daemon processes (i.e. processes that run continually in the background such as **cron**, **openup**, and **lpsched**. It also performs other initialization and housekeeping functions).
- restor** A Sys3 program used to restore individual files or entire file systems from a backup tape created with the Sys3 utility **/etc/dump**.
- setmnt** A program which cleans up an inaccurate mount table **mnttab** file. It is normally executed from the **rc** file but can also be directly invoked from the shell.

---

<b>shutdown</b>	A shell command which is used to bring the system down gracefully from multi-user to single-user state.
<b>ttytype</b>	A text file which identifies which TERM type corresponds to which tty port. The information in <b>ttytype</b> is used by <b>tset</b> and <b>login</b> to initialize the TERM variable at login time.
<b>umount</b>	A program which detaches ( <b>umounts</b> ) mounted file system(s) from Sys5 root directories.
<b>utmp</b>	A data file accessed by the <b>who</b> command. <b>utmp</b> keeps track of who is currently logged in to the system.
<b>wall</b>	(write all) A program that sends a broadcast message to all users. It is used by root to notify users that the system is going down, or other messages when immediate and disruptive notification is required.

---

## **/usr Directory**

The **/usr** directory is the only standard Sys5 directory in the root file system which contains extensive subdirectories. The **/usr** directory contains many directories and files used to support other commands and programs on the system. Users interact with the system by the use of shell commands which are located in the **/bin** directory and **/usr/bin** directory. Some of these commands use other files to store data or to execute independently to do their job.

Some of its subdirectories (**/usr/bin**, **/usr/plx**, **/usr/games**) contain more user-invoked programs. About half of the Sys5 shell commands reside in **/bin**. The other half reside in **/usr/bin**. The subdirectories under the **/usr** directory are described below.

**adm** This is the directory where all of the data collection files are kept for the system accounting function. Information contained in these files is used for generating system reports through the use of accounting programs in **/usr/lib/acct**. System accounting simply collects data and provides reports on the activity of your system (i.e. who logs on, how long, what commands are performed, the CPU time involved, etc.).

**bin** This is the second major Sys5 user command directory where about half of the Sys5 commands reside. The commands and programs available to users in **/usr/bin** are executable simple files. Programs in **/usr/bin** are available for general use by everyone. As the system administrator, you should not add programs to **/usr/bin**. Instead, you should set up your own command directory in **/usr** (i.e. **/usr/local** or **/usr/local/mybin**). Setting up your own command directory in **/usr** not only makes it easier to upgrade to a new release but also keeps programs which are not part of the AT&T release separate.

**games** This is the directory containing various game programs that come with your system.

**include** This directory contains all the include text files used in C programs.

- 
- lib** This directory contains a wide variety of files and archives which serve as libraries and support files for Sys5 programs and commands. Included are support files which are used by compilers, **troff/nroff** (text formatter), accounting, **cron**, **uucp**, **mailx**, and SCCS.
- mail** This directory contains files by username containing electronic mail messages sent/read by the standard Sys5 command, **mailx**. Otherwise, it is empty.
- man** This directory contains all the system manuals. It is an on-line version of the following manuals: *Sys5 UNIX User's Reference Manual*, *Sys5 UNIX Programmer's Reference Manual* and *Sys5 UNIX Administrator's Reference Manual*. Loading of this directory is optional.
- news** This directory contains files that are accessible by users via the **news** program. If you create a file in the **/usr/news** directory with a message relevant to all users on your system, users will be notified at login which news files are available for reading. See **news(1)** in the *Sys5 UNIX User's Reference Manual*.
- plx** A directory containing more Sys5 commands. These commands are either programs written at Plexus (such as **tape**, **copytape**, **dial**) or are part of the UC Berkeley enhancements package. These commands are included in Plexus' release of Sys5 to provide more capabilities.
- pub** A public directory containing ASCII text files and tables of general interest.
- spool** A directory containing spooling devices, buffers, and associated commands.
- src** A directory containing certain source code files and libraries used to reconfigure the operating system file **/unix**.
- tmp** An alternate scratchpad directory (like **/tmp**) used by some of the compilers and third party applications programs. It is usually empty.

---

The major Sys5 directories described in this chapter all come with the system as delivered from Plexus. These major Sys5 directories reside on the root file system part of the disk.

The disk can be divided into separate logical disks. Sys5 treats these logical disks as if they were separate physical disk drives. Depending on the storage capacity of the disk, you can make up to 15 separate file systems on a single physical disk drive. For example, you might want to store your user directories and files on one file system, your application programs on another. As you create alternate file systems, you will also want to create an empty directory in the root file system that you use to attach (mount) the contents stored on one of the file systems, so that those directories and files become a part of the overall Sys5 hierarchy. Later chapters will explain in detail how to set up file systems.

## **/stand Directory**

Programs in the **/stand** directory are stand-alone programs. That is, these programs are run independently of Sys5. They are run at the boot level prompt ( : ).

After pressing RESET, the system performs a self-test. Then, the boot prompt ( : ) displays prior to loading **/unix**. To invoke a stand-alone program at the boot prompt, you must specify the device and location of the stand-alone program you want to run. A listing of the **/stand** directory and a description of the stand-alone programs follows. Instructions explaining how to invoke stand-alone programs are given in the chapter *Stand-alone Programs*.

---

Enter the following command and a listing similar to Figure 3-5 (which follows) is displayed:

**ls -l /stand**

```
-rwxrwxr-x 1 bin bin 42474 Feb 16 12:00 cat
-rwxrwxr-x 1 bin bin 42632 Apr 22 10:26 dconfig
-rwxrwxr-x 1 bin bin 42812 Feb 16 12:00 dd
-rwxrwxr-x 1 bin bin 66178 Apr 2 16:01 dformat
-rwxrwxr-x 1 bin bin 40942 Feb 16 12:00 du
-rwxrwxr-x 1 bin bin 55842 Feb 16 12:00 fbackup
-rwxrwxr-x 1 bin bin 62720 Feb 16 12:00 fsck
-rwxrwxr-x 1 bin bin 53482 Feb 16 12:00 fsdb
-rwxrwxr-x 1 bin bin 52964 Feb 16 12:00 help
-rwxrwxr-x 1 bin bin 50748 Feb 16 12:00 ls
-rwxrwxr-x 1 bin bin 46392 Feb 16 12:00 mkfs
-rwxrwxr-x 1 bin bin 42308 Feb 16 12:00 od
-rwxrwxr-x 1 bin bin 49544 Feb 16 12:00 restor
```

**Figure 3-5.** Listing (ls -l) of the /stand Directory (Multibus)

---

Following is a description of the stand-alone programs' functions. Most of the stand-alone commands are described in the *Sys5 UNIX Administrator's Reference Manual*.

- cat** †            Concatenates and prints files.
- dconfig** ‡        Configures a disk with initialization information and the logical file system layout for Sys5.
- dd** †             Device-to-device copy program. Can read out of file systems.
- dformat** ‡        Disk format. Formats a disk drive, prompts for format information, and enables sparing of bad disk sectors.
- du** †             Reports disk usage.
- fbackup** ‡        Fast backup. Does a quick image backup from disk to tape, or restore from tape to disk. **fbackup** is not available on the P/90 or P/95.
- fsck** ‡            Interactive file system consistency check.
- fsdb** ‡            File system debugger.
- help** † ‡         Gives information about the release tape and how to use stand-alone programs.
- ls** †             Lists contents of directories.
- mkfs** ‡            Makes file system. Creates and organizes a bare file system.
- od** †             Octal dump. Dumps a file to the screen in octal or hex or ascii format.
- restor** †         Restores a file system from dump tape.

---

† See the *Sys5 UNIX User's Reference Manual*.

‡ See the *Sys5 UNIX Administrator's Reference Manual*.



---

## When To Run Stand-Alone Programs

The capability to run programs independently from Sys5 is sometimes required to create, alter, or repair the environment in which Sys5 runs. Under certain circumstances, you could not change the Sys5 environment without disabling the operating system kernel (and your accumulated files and programs) at the same time. Programs of this category alter or maintain the organization and/or format of the disk drive(s).

Some Sys5 stand-alone utilities are provided, not to alter the environment, but to check system status of various parameters while the system is down. Others provide device-to-device copying, backup, and/or restore capabilities when it is inconvenient or impossible to bring the system up to single-user or multi-user mode first. All of these stand-alone programs with the exception of **fbackup** (not available on the P/90 or P/95), **help**, and **dformat** are also available as regular Sys5 commands which can be invoked if the system is booted successfully.

Although you will frequently run the Sys5 versions of several stand-alone programs described in this chapter, for some programs you will rarely find it necessary to invoke stand-alone programs, unless it is not possible to boot the system.

You will occasionally need to invoke stand-alone programs if you:

- Install a new disk
- Increase the swap area or change its location
- Spare some bad disk tracks
- Implement more than two file systems on your disk
- Change the default logical disk sector addresses on the disk

Note that most of these functions are performed occasionally and are performed in single-user mode while running **/unix**. Additional information about stand-alone programs is described in the chapter *Stand-alone Programs*.

# STARTUP AND SHUTDOWN

---

This chapter describes:

- how to connect the system console
- how to start up and log in to your Plexus system
- how to shut down your Plexus system
- how to change system states

In the last section describing system states, an explanation describing what actions the system takes when the system is brought up for the first time (single-user mode — init state s) and when the system changes from single-user mode (init state s) to multi-user mode (init state 2) is presented. The procedures described in this chapter assume that the computer is installed correctly and all parts are functioning correctly as described in your Installation Guide.

---

## Connecting the System Console

To connect the system console to your Plexus system, refer to the pinout information in Table 4-1 for your cable.

**Table 4-1. Console Port Pinouts**

<b>Signal Name</b>	<b>Computer Console Port DB-25</b>	<b>Console Terminal DB-25</b>	<b>Signal Name</b>
RXD	2	2	TXD
TXD	3	3	RXD
GND	7	7	GND

## Normal Startup

When you turn the power on or press the RESET button, the system automatically begins its startup routine. It completes its self-test, then waits for the command to boot the operating system. When the operating system is booted (loaded from disk), the system comes up to single-user mode. The system can be used in single-user mode or brought to multi-user mode, init state 2. A step-by-step procedure is provided in the section *Startup, Login, and Shutdown Procedures*.

## PROM Self-test and Diagnostics

The system self-test program activates immediately after the system is powered on or reset. When it has completed its tests, the following message appears on the console:

```
PLEXUS SELFTEST REVX.X COMPLETE
```

Partial completion of this message indicates self-test failure and a hardware problem. Typically, the words PLEXUS SELFTEST REVX.X appear quickly and the word COMPLETE takes longer.

The system contains a bootstrap PROM that monitors system performance during the bootstrap procedure. If the system cannot complete the bootstrap procedure, it returns an error message either directly before or after the message, /unix.

---

## Startup, Login, and Shutdown Procedures

A step-by-step procedure of the startup sequence appears below. Both the visual display and necessary actions are displayed. Depending on your system configuration, your display might differ slightly. The P/55 display is shown below.

1. Turn your new Plexus system on by turning on the key. A message displays when the system has passed self-test:

```
PLEXUS SELFTTEST REVX.X COMPLETE
```

The CPU checks each board in the system and displays the system configuration:

```
SYSTEM CONFIGURATION im ac0 ac1 ex0 x.xMb
```

Where: im is the IMSP  
ac0 is ACP0  
ac1 is ACP1  
ex0 is the ethernet controller 0  
x.xMb is memory size

**[ NOTE ]** Your systems' configuration display may differ depending on the boards in your system.

Additional boards that might be displayed include:

Communication Processors:

vcX is VCP number X (P/90,P/95);

icX is ICP number X (P/60)

Tape Controllers:

ptX (P/35,P/55,P/75 cart. tape)

qcX (P/95 cart. tape)

stX (P/90 cart. tape)

rm (any system with reel-to-reel tape)

---

Disk Controllers:

pd/xsy zzz (IMSP)  
xy/xsy zzz (EMSP)  
sm/xsy zzz (AMSP)  
sd/xsy zzz MB host=*n* sid=*n* lun=*n* size=*nnnnnn* (MMSP)

Where: *x* = the number of the physical disk  
*y* = the number of the logical disk  
*zzz* = disk drive capacity  
host = host adapter number  
sid = SCSI ID  
lun = logic unit number  
size = number of 512-byte sectors

If a board in your system does not appear on the display, shut the system down (see the section *Shutting Down the System* in this chapter) and check the board. With the P/90, the SCSI host adapter board will not be displayed. For further information on self-test and diagnostics, see your system's Maintenance Manual.

Next, the system displays the boot prompt revision followed by the boot prompt ( : ).

PLEXUS PRIMARY BOOT REVX.X

:

2. At the boot prompt ( : ), press a carriage return and the operating system kernel, **/unix**, will load.

**[ NOTE ]** From the boot level prompt ( : ), you can also execute stand-alone programs. To avoid system failure, read the chapter *Stand-alone Programs* before booting any program other than **/unix**.

The system displays:

**/unix**

---

A short pause and the display continues:

```
Sys5/X.X.:sysX.Xxxx  
real mem = nnnnnnn  
avail mem = nnnnnnn  
INIT: SINGLE USER MODE
```

Where: *X.X.* is the revision level (such as 5.2)  
*xxx* is the number of users

When the root prompt displays ( # ), the system is in single-user mode.

3. At the # prompt, enter the following command to change to multi-user mode (init state 2) and send logins to terminals.

```
init 2
```

The system returns:

```
INIT: New run level: 2
```

4. Next, the system asks:

```
Is the date <date> correct? (y or n)
```

If date is correct, enter **y**. If the date is not correct, enter **n** and the correct date in the format *mmddhhmm[yy]*, where: *mm* is month, *dd* is day, *hh* is hour (24 hour clock), the second *mm* is minutes, and *yy* is the year (optional). These must be two digit numbers. See **date(1)** in the *Sys5 UNIX User's Reference Manual*.

5. The system then displays:

```
Do you want to check the file systems? (y or n)
```

Enter **y**. This executes the file system consistency check program, **fsck**. **fsck** checks all file systems listed in the file */etc/checklist*. We recommend that you ensure file system integrity with **fsck** before using the system in multi-user mode. Plexus recommends checking the file systems every time you reboot your system. The program **fsck(1M)** is described in the *Sys5 UNIX Administrator's Guide* in the chapter *File System Checking* and in the *Sys5 UNIX Administrator's Reference Manual*.

---

After executing **fsck**, the system displays a message similar to the following:

```
Initializing IMSP (if IMSP installed)
cron started
initializing ICP0 (if ICP0 installed)
initializing ACP0 (if ACP0 installed)
initializing VCP0 (if VCP0 installed — VMEbus)
```

Then the login prompt appears:

```
Console Login:
```

The system is now in the multi-user state.

If the startup sequence fails, see the chapter, *Crisis Recovery* for corrective action.

## Logging In

Plexus systems are shipped with a few active accounts, one of which is **root**. You must create additional user accounts. To log in, at the Console Login: prompt enter:

```
root
```

The system responds by returning some basic messages regarding the revision level of the operating system software, followed by the root or superuser prompt, #.

**[ NOTE ]** Root is the super user account. Normal permissions are ignored. Therefore, what you enter is carried out. BE CAREFUL!!!

---

## Setting Up System Passwords

Initially, the root account has no password. The root password gives read/write/execute permission on any file, directory, or program on your system. Obviously, root, as well as other system accounts, should have a password. Plexus Computers recommends that you add a password to the root account now. At the # prompt enter:

```
passwd root
```

To ensure a correct password entry, you are prompted twice to enter the password you have selected. The password must have at least six characters and must contain at least two alphabetic (upper or lower case) letters and at least one numeric or special character. The password you enter will not display on the screen.

You should also set up a password for all system accounts except sync. For a list of valid system accounts, enter:

```
cat /etc/passwd
```

The syntax of setting up a password for a system account is:

```
passwd sysacct
```

For example, to enter a password for the bin account, enter:

```
passwd bin
```

The procedure is the same as creating a password for root. See the *Sys5 UNIX User's Reference Manual* (**passwd(1)**) and the *Sys5 UNIX Programmer's Reference Manual* (**passwd(4)**) for more information on **passwd**.

## Logging Out

Log out by entering <CTRL d>. (Hold down the CTRL key and press the d key simultaneously—like using the shift key).



---

## Shutting Down the System

Shutting down the system means changing from multi-user state to single-user state. Your Plexus system **MUST** be shut down by executing the **/etc/shutdown** program.

**[ NOTE ]** If you have disks mirrored and are running Sys5, Release 1.5 or later, you do not have to perform any special procedures to shutdown your system. The **/etc/shutdown** procedure described in this section is adequate.

To shut the system down, follow the step-by-step procedure described below:

1. Be sure you are logged in as **root**. You should be on the system console. If you are not on the console, Sys5 will change the console to the port that **/etc/shutdown** is being run from.
2. Execute the **/etc/shutdown** program. Enter:

**/etc/shutdown X**

Where: *X* = seconds of grace period (default = 60 sec).

The system asks if you wish to send your own shutdown message if there are other users logged in. If you wish to enter your own message, respond affirmatively and enter your message. Conclude by entering a <CTRL d> on a line by itself.

The program **/etc/shutdown** sends a series of messages to all active terminals, warning that the system will shut down in *X* seconds (the length of the grace period mentioned above). This program also determines all active processes, kills all active processes, and flushes the buffers to disk.

- 
3. This takes several minutes and you will be prompted for yes/no responses when necessary. When asked the following:

Busy out (push down) the appropriate  
phone lines for this system.  
Do you want to continue ? (y or n)

Hang up any modems and enter:

**y**

The system displays the following:

All currently running processes will now be killed.  
Wait for INIT SINGLE USER MODE before halting.

```
INIT: New run level s
INIT: SINGLE USER MODE
#
```

**[ NOTE ]** Do not enter anything until the phrase INIT: SINGLE USER MODE appears followed by the # prompt. This might take several seconds.

4. Update the super-blocks and flush the system buffers by entering:

**sync**

The system is now in single-user mode. It may be used in single-user mode, restarted by pressing the RESET button, turning the keyswitch, entering **sys reset** (systems with the 68020 microprocessor only), or it may be turned off by switching the system power switch to the OFF position.

It is advisable to run **fsck** to check the integrity of your file systems prior to powering the system off.

---

## System States

Read the remainder of this chapter if you want to know more about system states on your Plexus system. The remainder of this chapter describes what actions the system takes when:

- the system is brought up for the first time
- the system changes from single-user mode (init state s) to multi-user mode (init state 2)

## Entering Single-User Mode

Towards the end of the system bring up procedure, Sys5 runs the program `/etc/init`. `/etc/init`, an executable program in binary format, is responsible for the events that follow, including creating the super-user shell.

The `init` process looks in the `/etc/inittab` file for a default entry listed below:

```
is:s:initdefault:
```

Where: `is` is one or two characters used to uniquely identify an entry.

`s` is the run-level (i.e., system state). In this case, `init s` (single-user mode) is specified. `initdefault` is the action field. The `initdefault` action is scanned when `init` is initially invoked (when the system is brought up for the first time). `init` uses this entry to determine which run level to enter. `init` takes the highest run level specified in the run level field and uses that as its initial state.

Because only one run level (run level `s`) was specified, the system is brought up in single-user mode (init state `s`). If `init` does not find an `initdefault` entry in `/etc/inittab`, it requests an initial run level from the system console at boot time.

---

## Entering Multi-User Mode

When you first bring up your system, you will be in single-user mode (init state s) because of the `initdefault` entry in `/etc/inittab`. You change from single-user mode (init state s) to multi-user mode (init state 2) by entering:

```
init 2
```

When you enter `init 2`, the `init` process selects lines in the `/etc/inittab` file which specify run level 2. The `/etc/rc` file is called by `init`. This file is a shell script and lists processes to be performed. Some of the tasks the system performs when going to multi-user mode include mounting file systems and starting processes which run in the background, such as `cron`.

On the following pages are sample listings of the `/etc/rc` file. This file is different for systems with the Motorola 68000 microprocessor (Multibus-based system), and systems with the Motorola 68020 microprocessor with either the Multibus or VMEbus-based systems. There is a listing for each type of system, and a somewhat detailed explanation of sections of the `/etc/rc` file for the Motorola 68020 Multibus-based system (Version S1.7 of UNIX).

Lines beginning with a `#` are comments.

---

Figure 4-1, following, is a listing of the `/etc/rc` file for Motorola 68000 systems, Version M1.7, Multibus systems.

```
# @#Mrc.sh      7.2

export TZ
TZ = `cat /etc/TIMEZONE`
if [ ! -f /etc/mnttab ] ; then
    /etc/devnm / | grep -v swap | grep -v root | /etc/setmnt
fi
set `who -r`
CurrentState=$7
TimesInCurrentState=$8
PreviousState=$9

case $CurrentState in
s|1)
    if [ $TimesInCurrentState -ne 0 ] ; then
        # put umounts here
        sync
    fi
    ;;
2)
    if [ $TimesInCurrentState -eq 0 ] ; then
        sync
        sleep 3          # time for the above sync to happen
        /etc/dnld -dd -a 5000000 -f /usr/lib/dnld/imsc -o /dev/im0
    fi
    # put mounts here (/usr, etc.)
    # Uncomment the following to have /lib mounted as a read
    # only ram disk.
    # This may require installing some software products in
    # single user mode
    #
    #SIZE=`du -s /lib | awk `{ print $1 }`
    #SIZE=`expr \( $SIZE / 20 \) + $SIZE`
    #if /usr/plx/ramdisk /dev/rram/a ${SIZE}k ; then
    # echo "Allocated ${SIZE} kilobytes of ram for /lib"
    # mkfs -q /dev/rram/a $SIZE
```

```

# mount /dev/ram/a /tmp >/dev/null
# (cd /lib ; find . -depth -print | cpio -pdm /tmp >/dev/null)
# umount /dev/ram/a
# /dev/ram/a /lib -r
# echo "Mounted /lib as a read only ramdisk"
#else
# echo "Did not make ramdisk for /lib"
#
#fi
#
# Uncomment the following to have /bin mounted as a read
# only ram disk. This will improve the start up time for
# all commands in /bin.
# This may require installing some software products in
# single user mode
#
#SIZE=`du -s /bin | awk `{ print $1 }`
#SIZE=`expr \( $SIZE / 20 \) + $SIZE`
#if /usr/plx/ramdisk /dev/rram/b ${SIZE}k ; then
# echo "Allocated ${SIZE} kilobytes of ram for /bin"
# mkfs -q /dev/rram/b $SIZE
# mount /dev/ram/b /tmp >/dev/null
# (cd /bin ; find . -depth -print | cpio -pdm /tmp >/dev/null)
# umount /dev/ram/b
# mount /dev/ram/b /bin -r
# echo "Mounted /bin as a read only ramdisk"
#else
# echo "Did not make ramdisk for /bin"
#fi
#
#
# Uncomment the following and give a reasonable value
# to USRTMPSZ to have /usr/tmp be a ramdisk. USRTMPSZ
# is the amount of memory allocated, in megabytes.
# This technique could also be used for /tmp, however
# ex.preserve would not be able to retrieve temporary
# files that were being edited when the system was
# shutdown.
#

```

```

#USRTMPSZ=1
#if /usr/plx/ramdisk /dev/rram/c ${USRTMPSZ}m ; then
#   echo "Allocated ${USRTMPSZ} megabytes of ram for /usr/tmp"
#   mkfs -q /dev/rram/c `expr ${USRTMPSZ} \* 1024`
#   mount /dev/ram/c /usr/tmp
#   echo "Mounted /usr/tmp as a read/write ramdisk"
#else
#   echo "Did not make ramdisk for /usr/tmp"
#fi

#
#   Uncomment the following to have a directory of executables in
#   ram.  Add any files you want in ram to the definition of $FAST
#   The directory is called /fast.  This technique
#   provides the most flexibility in what goes into ram, but to
#   be effective all users must have /fast added to their shell
#   path.  If /bin is being loaded into ram, it is simpler to
#   link all fast executables there.  Then there is no need to
#   change the users environment
#
#FAST="\
#/usr/bin/fgrep \
#/usr/bin/lp \
#/usr/bin/mailx \
#/usr/bin/vi \
#"
#FASTSZ=0
#for f in $FAST ; do
#   SIZE=`du -s $f | awk `{ print $1 }`
#   FASTSZ=`expr $FASTSZ + $SIZE`
#done
#FASTSZ=`expr \( $FASTSZ / 20 \) + $FASTSZ`
#if /usr/plx/ramdisk /dev/rram/d ${FASTSZ}k ; then
#   echo Allocated $FASTSZ kilobytes of ram for /fast
#   if [ ! -d /fast ]; then mkdir /fast ; fi
#   mkfs -q /dev/rram/d $FASTSZ
#   mount /dev/ram/d /fast
#   cp $FAST /fast
#   ln /fast/vi /fast/ex

```

```

# ln /fast/vi /fast/view
# echo "Mounted /fast as a read/write ramdisk"
#else
# echo "Did not make ramdisk for /fast"
#fi

if [ $TimesInCurrentState -eq 0 ] ; then
    /usr/lib/ex3.9preserve -
    rm -f /tmp/*
fi
# put chowns for uucp L-devices DIR devices here, i.e.
# chown uucp /dev/tty0
rm -f /usr/spool/uucp/LCK*
rm -f /usr/adm/acct/nite/lock*
mv /usr/adm/sulog /usr/adm/OLDSulog
mv /usr/lib/cron/log /usr/lib/cron/OLDlog
> /usr/lib/cron/log
/etc/cron
echo cron started
# /bin/su - adm -c /usr/lib/acct/startup
# echo process accounting started
# rm -f /usr/spool/lp/SCHEDLOCK
# /usr/lib/lpsched
# echo "LP scheduler started"
#/usr/lib/errdemon
#echo errdemon started
# /bin/su - sys -c "/usr/lib/sa/sadc /usr/adm/sa/sa`date +%d`"
if [ $TimesInCurrentState -eq 0 ] ; then
# ICP's (if any) are first starting at number zero and incrementing
# upward by one. Then ACP's (if any) numbered starting with one mo
# than the last ICP number and incrementing upward by one.
# example: 1 ICP and 2 ACP's
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000
# /etc/dnld -L -da 1 2 -f /usr/lib/dnld/acp -a 2000
# example: 3 ICP's and 1 ACP
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic1 -a 4000
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic2 -a 4000
# /etc/dnld -L -da 3 1 -f /usr/lib/dnld/acp -a 2000

```



```

#      example:  0 ICP's and 4 ACP's
#      /etc/dnld -L -da 0 4 -f /usr/lib/dnld/acp -a 2000
#      example:  2 ICP's and 0 ACP's
#      /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000
#      /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic1 -a 4000
#      Uncommented the following lines that are appropriate for your
#      configuration.
#      /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000
#      /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic1 -a 4000
#      /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic2 -a 4000
#      /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic3 -a 4000
#      /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic4 -a 4000
# comment:      -da <starting acp #> <# of acp's>
#      /etc/dnld -L -da ? ? -f /usr/lib/dnld/acp -a 2000
#      This section of commands are for specifying modem ports on ACP's.
#      Remember, on ICP's, hardware jumpers need to be changed to
#      select a port to be either local or modem.  For ACP's, a command
#      line must be placed here to inform the ACP that a port is
#      connected to a modem or it will default to local mode.
#      (ex.      stty -clocal </dev/tty##)
#              sleep 3
#              echo Configure Modem Ports
#      stty -clocal </dev/tty##
#
fi
# rm -f /usr/spool/lp/SCHEDLOCK
# /usr/lib/lpshut >>/dev/null 2>>/dev/null
# /usr/lib/lpsched
# echo "LP scheduler started"
# sleep 1
;;
esac

```

Figure 4-1. Sample /etc/rc File, Motorola 68000, Multibus (M1.7)

---

Figure 4-2, following, is a listing of the `/etc/rc` file for Motorola 68020 systems, Version S1.7, Multibus systems.

```
#  @(#)Src.sh  7.2

export TZ
TZ=`cat /etc/TIMEZONE`
if [ ! -f /etc/mnttab ] ; then
    /etc/devnm / | grep -v swap | grep -v root | /etc/setmnt
fi
set `who -r`
CurrentState=$7
TimesInCurrentState=$8
PreviousState=$9

case $CurrentState in
S|1)
    if [ $TimesInCurrentState -ne 0 ] ; then
        # put umounts here
        sync
    fi
    if sys keyoff || sys hot      # this is a complete shutdown
    then
        echo "Unloading tape(s)."
        /usr/plx/tape -f /dev/rrm/0m unload
        sleep 15
        sync; sync; sync
        echo "Graceful shutdown complete."
        echo "System power off...  \c"
        sleep 5
        sys off
    fi
    ;;
2)
    /etc/init c # check for active ccb lines
    if [ $TimesInCurrentState -eq 0 ] ; then
        sync
        sleep 3      # time for the above sync to happen
        /etc/dnld -dd -a 5000000 -f /usr/lib/dnld/imsc -o /dev/im0
    fi
```

```

# put mounts here (/usr, etc.)
#
# Uncomment the following to have /lib mounted as a read
# only ram disk.
# This may require installing some software products in
# single user mode
#
#SIZE=`du -s /lib | awk '{ print $1 }'`
#SIZE=`expr \( $SIZE / 20 \) + $SIZE`
#if /usr/plx/ramdisk /dev/rram/a ${SIZE}k ; then
# echo "Allocated ${SIZE} kilobytes of ram for /lib"
# mkfs -q /dev/rram/a $SIZE
# mount /dev/ram/a /tmp >/dev/null
# (cd /lib ; find . -depth -print | cpio -pdm /tmp >/dev/null)
# umount /dev/ram/a
# /dev/ram/a /lib -r
# echo "Mounted /lib as a read only ramdisk"
#else
# echo "Did not make ramdisk for /lib"
#
#fi
#
# Uncomment the following to have /bin mounted as a read
# only ram disk. This will improve the start up time for
# all commands in /bin.
# This may require installing some software products in
# single user mode
#
#SIZE=`du -s /bin | awk '{ print $1 }'`
#SIZE=`expr \( $SIZE / 20 \) + $SIZE`
#if /usr/plx/ramdisk /dev/rram/b ${SIZE}k ; then
# echo "Allocated ${SIZE} kilobytes of ram for /bin"
# mkfs -q /dev/rram/b $SIZE
# mount /dev/ram/b /tmp >/dev/null
# (cd /bin ; find . -depth -print | cpio -pdm /tmp >/dev/null)
# umount /dev/ram/b
# mount /dev/ram/b /bin -r
# echo "Mounted /bin as a read only ramdisk"
#else

```

```

# echo "Did not make ramdisk for /bin"
#fi

#
# Uncomment the following and give a reasonable value
# to USRTMPSZ to have /usr/tmp be a ramdisk. USRTMPSZ
# is the amount of memory allocated, in megabytes.
# This technique could also be used for /tmp, however
# ex.preserve would not be able to retrieve temporary
# files that were being edited when the system was
# shutdown.
#
#USRTMPSZ=1
#if /usr/plx/ramdisk /dev/rram/c ${USRTMPSZ}m ; then
# echo "Allocated ${USRTMPSZ} megabytes of ram for /usr/tmp"
# mkfs -q /dev/rram/c `expr ${USRTMPSZ} \* 1024`
# mount /dev/ram/c /usr/tmp
# echo "Mounted /usr/tmp as a read/write ramdisk"
#else
# echo "Did not make ramdisk for /usr/tmp"
#fi

#
# Uncomment the following to have a directory of executables in
# ram. Add any files you want in ram to the definition of $FAS
# The directory is called /fast. This technique
# provides the most flexibility in what goes into ram, but to
# be effective all users must have /fast added to their shell
# path. If /bin is being loaded into ram, it is simpler to
# link all fast executables there. Then there is no need to
# change the users environment
#
#FAST="\
#/usr/bin/fgrep \
#/usr/bin/lp \
#/usr/bin/mailx \
#/usr/bin/vi \
#"
#FASTSZ=0

```

```

#for f in $FAST ; do
#   SIZE=`du -s $f | awk `{ print $1 }`
#   FASTSZ=`expr $FASTSZ + $SIZE`
#done
#FASTSZ=`expr \( $FASTSZ / 20 \) + $FASTSZ`
#if /usr/plx/ramdisk /dev/rram/d ${FASTSZ}k ; then
#   echo Allocated $FASTSZ kilobytes of ram for /fast
#   if [ ! -d /fast ]; then mkdir /fast ; fi
#   mkfs -q /dev/rram/d $FASTSZ
#   mount /dev/ram/d /fast
#   cp $FAST /fast
#   ln /fast/vi /fast/ex
#   ln /fast/vi /fast/view
#   echo "Mounted /fast as a read/write ramdisk"
#else
#   echo "Did not make ramdisk for /fast"
#fi

if [ $TimesInCurrentState -eq 0 ] ; then
    /usr/lib/ex3.9preserve -
rm -f /tmp/*
fi
# put chowns for uucp L-devices DIR devices here, i.e.
# chown uucp /dev/tty0
rm -f /usr/spool/uucp/LCK*
rm -f /usr/adm/acct/nite/lock*
mv /usr/adm/sulog /usr/adm/OLDSulog
mv /usr/lib/cron/log /usr/lib/cron/OLDlog
> /usr/lib/cron/log
/etc/cron
echo cron started
# /bin/su - adm -c /usr/lib/acct/startup
# echo process accounting started
# /usr/lib/errdemon
# echo errdemon started
# /bin/su - sys -c "/usr/lib/sa/sadc /usr/adm/sa/sa`date +%d`"
if [ $TimesInCurrentState -eq 0 ] ; then
#
# ICP's (if any) are first starting at number zero and incrementing

```

```

# upward by one. Then ACP's (if any) numbered starting with one more
# than the last ICP number and incrementing upward by one.
# example: 1 ICP and 2 ACP's
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000
# /etc/dnld -L -da 1 2 -f /usr/lib/dnld/acpASY -a 2000
# example: 3 ICP's and 1 ACP
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic1 -a 4000
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic2 -a 4000
# /etc/dnld -L -da 3 1 -f /usr/lib/dnld/acpASY -a 2000
# example: 0 ICP's and 4 ACP's
# /etc/dnld -L -da 0 4 -f /usr/lib/dnld/acpASY -a 2000
# example: 2 ICP's and 0 ACP's
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic1 -a 4000
# Uncommented the following lines that are appropriate for your
# configuration.
#
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic1 -a 4000
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic2 -a 4000
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic3 -a 4000
# /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic4 -a 4000
# Comment: -da <starting acp #> <# of acp's>
# /etc/dnld -L -da ? ? -f /usr/lib/dnld/acpASY -a 2000
#
# This section of commands are for specifying modem ports on ACP's.
# Remember, on ICP's, hardware jumpers need to be changed to
# select a port to be either local or modem. For ACP's, a command
# line must be placed here to inform the ACP that a port is
# connected to a modem or it will default to local mode.
# (ex. stty -clocal </dev/tty##>
# sleep 3
# echo Configure Modem Ports
# stty -clocal </dev/tty##>
# make sure UNIX knows we're in init 2
# /etc/init 2
fi
# rm -f /usr/spool/lp/SCHEDLOCK

```

```

# /usr/lib/lpshut >>/dev/null 2>>/dev/null
# /usr/lib/lpsched
# echo "LP scheduler started"
# sleep 1
;;
esac

```

**Figure 4-2.** Sample `/etc/rc` File, Motorola 68020, Multibus (S1.7)

## Walk-through of `/etc/rc` File

An explanation of this version of the `/etc/rc` file is presented now. Following it is a listing of the `/etc/rc` file for the Motorola 68020 VMEbus-based system.

The S1.7 version of the `/etc/rc` file for Motorola 68020 Multibus-based systems will be explained a portion at a time.

```

export TZ
TZ = `cat /etc/TIMEZONE`

if [ ! -f /etc/mnttab ] ; then
    /etc/devnm / | grep -v swap | grep -v root | /etc/setmnt
fi
set `who -r`
CurrentState=$7
TimesInCurrentState=$8
PreviousState=$9

```

The `/etc/TIMEZONE` file, which contains time zone information for your system, is called. If the file `/etc/mnttab` does not exist, it is created. This file is initialized with information from the root file system. The system determines the current state by executing `who -r`. The variable `$7` is assigned to `CurrentState`, `$8` is assigned to `TimesInCurrentState`, and `$9` is assigned to `PreviousState`.

---

The next portion of the `/etc/rc` file follows, with its explanation.

```
case $CurrentState in
s|1)
    if [ $TimesInCurrentState -ne 0 ] ; then
        # put umounts here
        sync
    fi
```

The first line begins a case statement which tests for one of two conditions. The procedures in the first case statement are carried out if the system is in single-user mode (init S or init 1), signified by `s|1`. If either state S or state 1 is true, the directions in the first case statement are executed.

If the system is in state S or state 1, it tests if the number of times in the current state is equal to 0. If this is true, it unmounts (disattaches) the file systems listed under `# put umounts here`. Enter the **umount** commands to detach file systems you want unmounted when the system goes down below the line `# put umounts here` in the `/etc/rc` file. After unmounting file systems, the system performs a `sync`. The **fi** statement ends the **if** statement.

Continuing:

```
if sys keyoff || sys hot    # this is a complete shutdown
then
    echo "Unloading tape(s)."
```

```
    /usr/plx/tape -f /dev/rrm/0m unload
    sleep 15
    sync; sync; sync
    echo "Graceful shutdown complete."
    echo "System power off...  \c"
    sleep 5
    sys off
fi
;;
```

This section performs a graceful shut down of the system if the key is turned off or if the system gets too hot.



---

The `;;` ends the first case, **S|1**).

The explanation of the `/etc/rc` file continues.

2)

```
/etc/init c # check for active ccb lines
if [ $TimesInCurrentState -eq 0 ] ; then
    sync
    sleep 3    # time for the above sync to happen
    /etc/dnld -dd -a 5000000 -f /usr/lib/dnld/imsc -o /dev/im0
fi
# put mounts here (/usr, etc.)
```

If the system is in state 2, signified by `2`), the procedures in this second case are carried out. This case (multi-user mode, state 2) continues to the end of the `/etc/rc` file. If the system is in state 2 and if the number of times the system is in the current state is equal to 0, the system does the following:

- performs a `sync` and waits for three seconds.
- downloads the `imsc` with code to control the Intelligent Mass Storage Processor disk and tape controller.

The procedures listed below are performed every time the system enters multi-user mode (state 2). File systems listed under `# put mounts here (/usr, etc.)` are mounted (attached). Enter the file systems you want mounted when the system enters multi-user mode following the comment `# put mounts here (/usr, etc.)`.

Pages of examples of using `ramdisk` follow. Uncomment any section(s) that you want to use. See **ramdisk(1M)** in the *Sys5 UNIX Administrator's Reference Manual* for more information about **ramdisk**.

---

Following the pages of ramdisk information, the listing continues:

```
if [ $TimesInCurrentState -eq 0 ] ; then
    /usr/lib/ex3.9preserve -
rm -f /tmp/*
fi
```

If the number of times the system is in the current state (init state 2) is equal to zero, the system does the following:

- Finds all Ex\* or Rx\* files left in **/tmp** by the text editors **vi** or **ex** and reconstitutes these files into editable form. These files are put into the **/usr/preserve** directory. A **mail** message is sent to the owner telling the person that the file(s) can be recovered using the recover option with **vi** or **ex** (i.e. **vi -r filename** or **ex -r filename**).
- Removes files from **/tmp**.

The **/etc/rc** file continues:

```
# put chowns for uucp L-devices DIR devices here, .i.e
# chown uucp /dev/tty0
rm -f /usr/spool/uucp/LCK*
rm -f /usr/adm/acct/nite/lock*
mv /usr/adm/sulog /usr/adm/OLDSulog
mv /usr/lib/cron/log /usr/lib/cron/OLDlog
> /usr/lib/cron/log
/etc/cron
echo cron started
# /bin/su - adm -c /usr/lib/acct/startup
# echo process accounting started
# /usr/lib/errdemon
# echo errdemon started
# /bin/su - sys -c "/usr/lib/sa/sadc /usr/adm/sa/sa`date +%d`"
```

The procedures in this section do the following:

- Remove the lock files from the communications spooler, **uucp**.
- Remove lock files from accounting.

- Move the file `sulog` to `OLDsulog`.
- Move the file `/usr/lib/cron/log` to `R/usr/lib/cron/OLDlog`.
- Create an empty file named `/usr/lib/cron/log`.
- Start the program `cron` and display `cron` started on the system console.
- Start accounting and display process accounting started on the system console.
- Start the program `/etc/errdemon` and display `errdemon` started on the system console.
- Start the system activity report for the current day.

The explanation continues:

```

if [ $TimesInCurrentState -eq 0 ] ; then
#
#   ICP's (if any) are first starting at number zero and incrementing
#   upward by one. Then ACP's (if any) numbered starting with one more
#   than the last ICP number and incrementing upward by one.

```

Examples of mixtures of ICP's and ACP's follow and the listing continues:

```

#   Uncommented the following lines that are appropriate for your
#   configuration.
#
#       /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000
#       /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic1 -a 4000
#       /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic2 -a 4000
#       /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic3 -a 4000
#       /etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic4 -a 4000
#   Comment: -da <starting acp #> <# of acp's>
#       /etc/dnld -L -da ?? -f /usr/lib/dnld/acp -a 2000

```

If the system is in multi-user mode (init state 2) and the number of times in the current state is equal to 0, download ICPs and ACPs appropriate for your configuration. See the examples in the

---

`/etc/rc` file. Uncomment the appropriate download lines for ICPs. Edit the ACP download line and change ?? to the appropriate values following the examples in the `/etc/rc` file.

The last lines of the `/etc/rc` file with their explanation follow.

```
# This section of commands are for specifying modem ports on acp's.
# Remember, on rcp's, hardware jumpers need to be changed to
# select a port to be either local or modem. For acp's, a command
# line must be placed here to inform the acp that a port is
# connected to a modem or it will default to local mode.
# (ex. stty -clocal </dev/tty##)
# sleep 3
# echo Configure Modem Ports
# stty -clocal </dev/tty##
# # make sure UNIX knows we're in init 2
# /etc/init 2

fi

# rm -f /usr/spool/lp/SCHEDLOCK
# /usr/lib/lpshut >>/dev/null 2>>/dev/null
# /usr/lib/lpsched
# echo "LP scheduler started"
# sleep 1

;;

esac
```

The above instructions perform the following:

- The system performs a sleep for 3 seconds.
- If you have an ACP port connected to a modem, uncomment and edit the line `stty -clocal </dev/tty##`. In other words, remove the # from the beginning of the line and change ## to the appropriate tty number (such as tty4).
- `/etc/init 2` is invoked. This ensures that the system is in multi-user mode (init state 2).
- The `fi` statement ends the if statement, the `;;` statement ends the second case — the multi-user mode case.
- The `esac` statement ends the case statement.

---

Figure 4-3, following, is a listing of the `/etc/rc` file for Motorola 68020 systems, Version K1.7, VMEbus systems.

```
# @(#)Krc.sh 6.5

export TZ
TZ=`cat /etc/TIMEZONE`
if [ ! -f /etc/mnttab ] ; then
    /etc/devnm / | grep -v swap | grep -v root | /etc/setmnt
fi
set `who -r`
CurrentState=$7
TimesInCurrentState=$8
PreviousState=$9

case $CurrentState in
S|1)
    if [ $TimesInCurrentState -ne 0 ] ; then
        # put umounts here
        sync
    fi
    if sys keyoff || sys hot# this is a complete shutdown
    then
        echo "Unloading tape(s). "
        /usr/plx/tape -f /dev/rrm/0m unload
        sleep 15
        sync; sync; sync
        echo "Graceful shutdown complete."
        echo "System power off...  \c"
        sleep 5
        sys off
    fi
    ;;
2)
    /etc/init c # check for active ccb lines
    # put mounts here (/usr, etc.)

    #
    # Uncomment the following to have /lib mounted as a read
```

---

```

#   only ram disk.
#   This may require installing some software products in
#   single user mode
#
#SIZE=`du -s /lib | awk `{ print $1 }`
#SIZE=`expr \( $SIZE / 20 \) + $SIZE`
#if /usr/plx/ramdisk /dev/rram/a ${SIZE}k ; then
#   echo "Allocated ${SIZE} kilobytes of ram for /lib"
#   mkfs -q /dev/rram/a $SIZE
#   mount /dev/ram/a /tmp >/dev/null
#   (cd /lib ; find . -depth -print | cpio -pdm /tmp >/dev/null)
#   umount /dev/ram/a
#   mount /dev/ram/a /lib -r
#   echo "Mounted /lib as a read only ramdisk"
#else
#   echo "Did not make ramdisk for /lib"
#fi

#
#   Uncomment the following to have /bin mounted as a read
#   only ram disk. This will improve the start up time for
#   all commands in /bin.
#   This may require installing some software products in
#   single user mode
#
#SIZE=`du -s /bin | awk `{ print $1 }`
#SIZE=`expr $SIZE / 20 ) + $SIZE`
#if /usr/plx/ramdisk /dev/rram/b ${SIZE}k ; then
#   echo "Allocated ${SIZE} kilobytes of ram for /bin"
#   mkfs -q /dev/rram/b $SIZE
#   mount /dev/ram/b /tmp >/dev/null
#   (cd /bin ; find . -depth -print | cpio -pdm /tmp >/dev/null)
#   umount /dev/ram/b
#   mount /dev/ram/b /bin -r
#   echo "Mounted /bin as a read only ramdisk"
#else
#   echo "Did not make ramdisk for /bin"
#fi

```

```

#
# Uncomment the following and give a reasonable value
# to USRTMPSZ to have /usr/tmp be a ramdisk. USRTMPSZ
# is the amount of memory allocated, in megabytes.
# This technique could also be used for /tmp, however
# ex.preserve would not be able to retrieve temporary
# files that were being edited when the system was
# shutdown.
#
#USRTMPSZ=1
#if /usr/plx/ramdisk /dev/rram/c ${USRTMPSZ}m ; then
# echo "Allocated ${USRTMPSZ} megabytes of ram for /usr/tmp"
# mkfs -q /dev/rram/c `expr ${USRTMPSZ} 1024`
# mount /dev/ram/c /usr/tmp
# echo "Mounted /usr/tmp as a read/write ramdisk"
#else
# echo "Did not make ramdisk for /usr/tmp"
#fi

#
# Uncomment the following to have a directory of executables in
# ram. Add any files you want in ram to the definition of $FAST
# The directory is called /fast. This technique
# provides the most flexibility in what goes into ram, but to
# be effective all users must have /fast added to their shell
# path. If /bin is being loaded into ram, it is simpler to
# link all fast executables there. Then there is no need to
# change the users environment
#
#FAST="\
#/usr/bin/fgrep \
#/usr/bin/lp \
#/usr/bin/mailx \
#/usr/bin/vi \
#"
#FASTSZ=0
#for f in $FAST ; do
# SIZE=`du -s $f | awk '{ print $1 }'`

```

```

# FASTSZ=`expr $FASTSZ + $SIZE`
#done
#FASTSZ=`expr $FASTSZ / 20 ) + $FASTSZ`
#if /usr/plx/ramdisk /dev/rram/d ${FASTSZ}k ; then
# echo Allocated $FASTSZ kilobytes of ram for /fast
# if [ ! -d /fast ]; then mkdir /fast ; fi
# mkfs -q /dev/rram/d $FASTSZ
# mount /dev/ram/d /fast
# cp $FAST /fast
# ln /fast/vi /fast/ex
# ln /fast/vi /fast/view
# echo "Mounted /fast as a read/write ramdisk"
#else
# echo "Did not make ramdisk for /fast"
#fi

if [ $TimesInCurrentState -eq 0 ] ; then
    /usr/lib/ex3.9preserve -
    rm -f /tmp/*
fi
# put chowns for uucp L-devices DIR devices here, i.e.
# chown uucp /dev/tty0
rm -f /usr/spool/uucp/LCK*
rm -f /usr/adm/acct/nite/lock*
mv /usr/adm/sulog /usr/adm/OLDSulog
mv /usr/lib/cron/log /usr/lib/cron/OLDlog
> /usr/lib/cron/log
/etc/cron
echo cron started
# /bin/su - adm -c /usr/lib/acct/startup
# echo process accounting started
#/usr/lib/errdemon
#echo errdemon started
# /bin/su - sys -c "/usr/lib/sa/sadc /usr/adm/sa/sa`date +%d`"
if [ $TimesInCurrentState -eq 0 ] ; then
#
# All VCP's must be downloaded and configured during system bootup.
# Comment: -dv <starting vcp #> <# of vcp's>
# /etc/dataio -dv ?? -f /usr/lib/dnld/vcp5 -a 2000

```



```

# To downline load 1 VCP:
#   /etc/dataio -dv 0 1 -f /usr/lib/dnld/vcp5 -a 2000
# To downline load 3 VCP's:
#   /etc/dataio -dv 0 3 -f /usr/lib/dnld/vcp5 -a 2000
#   rm /etc/vconfig/log
# To configure asynchronous terminal lines use:
#   /etc/vconfig/vconfig -cl /etc/vconfig/ASYNC
# To configure parallel printer lines use:
#   /etc/vconfig/vconfig -cl /etc/vconfig/PARA
#
#
# This section of commands are for specifying modem ports on VCP's.
# For all VCP's, a command line must be placed here to inform the v
# that a port is connected to a modem or it will default to local m
# (ex.   stty -clocal </dev/tty##)
#       sleep 3
#       echo Configure Modem Ports
#       stty -clocal </dev/tty##
#       # make sure UNIX knows we're in init 2
#       /etc/init 2
#
fi
# rm -f /usr/spool/lp/SCHEDLOCK
# /usr/lib/lpshut >>/dev/null 2>>/dev/null
# /usr/lib/lpsched
# echo "LP scheduler started"
# sleep 1
;;
esac

```

**Figure 4-3.** Sample /etc/rc File, Motorola 68020, VMEbus, (K1.7)

# GETTING STARTED

---

This chapter describes:

- how to set up root's login environment.
- what system states Plexus provides and how to change system states (init 2 to s or s to 2).
- the autoboot feature, how to enable or disable autoboot as well as other init modes ( i.e. q ).
- how to modify **/etc/dnld** (Multibus systems) or **/etc/dataio** (P/90 and P/95 systems) commands in the **/etc/rc** file.
- how to prevent large files from being created (runaway processes)
- information about the Sys5 accounting programs
- how to use the reel-to-reel and cartridge tape drives.

---

## Setting Up Root's Login Environment

At this point, you have a system up with the root account active. The default shell environment for root, however, is not fully defined. For example, the default root account does not recognize commands in `/usr/plx`, and the system does not know specifically what kind of terminal your console is. By defining a few parameters at this point you can avoid typing lengthy pathnames and enjoy the benefits of a full screen editor. This will make subsequent setup procedures much easier. The examples that follow use the Bourne shell.

1. Specify your terminal to the system so you can use a full screen editor. If your terminal is a DEC\* VT-100 or equivalent, **TERM=vt100** is already defined in root's `/.profile` file. Therefore, you can skip this step. If your terminal is not a vt100, find a file in the `/usr/lib/terminfo` directory structure to match your console terminal. See the *Sys5 UNIX Programmer's Reference Manual* under **terminfo(4)**.

The **terminfo** directory contains many files. Each file provides configuration information to programs, such as `vi`, which look for information about a specific terminal. Because of the huge number of files, the **terminfo** directory is actually two layers deep to keep things manageable. The actual structure is `/usr/lib/terminfo/n/name` where *name* is the file describing your terminal and *n* is the first letter or number of *name*.

For example, the WYSE 100† terminal information is contained in:

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

---

\* DEC and VT100 are trademarks of Digital Equipment Corporation.

† WYSE and WYSE 100 are trademarks of WYSE Technologies, Inc.

---

After you have found your terminal code in `/usr/lib/terminfo/n/name`, specify it to the system.

For example, if your terminal is a WYSE 100 and you are running the Bourne shell (`/bin/sh`), at the `#` prompt enter:

```
TERM=wy100
export TERM
```

2. To list the all of the files (including the hidden dot files) in your current working directory enter:

```
ls -a
```

3. Using one of the editors, edit root's `./profile` file and modify the value of the `TERM` variable. To use `vi`, enter:

```
vi ./profile
```

Note that the file name of `./profile` begins with a period ( `.` ). In Sys5, files beginning with a period ( `.` ) are hidden unless you enter the `-a` option to `ls`.

4. Root's `./profile` file, which is invoked whenever root logs in, sets up root's login environment. The login environment includes default paths to be searched to find commands and default terminal characteristics. Change the value of the `TERM` variable as follows. For example, if you have a Wyse 100 terminal, substitute `TERM=vt100` with:

```
TERM=wy100
```

Following is root's `./profile` as shipped from Plexus:

---

```

stty erase '^h' kill '^x' echoe
PATH=./usr/plx:/bin:/usr/bin:/etc
TZ=PST8PDT
TERM=vt100
EXINIT='set ai nows redraw sm'
export PATH TZ TERM EXINIT
if /etc/sys autoboot && [ '/etc/sys initstate' -eq 0 ]
then      # autoboot set and init = 'S'
          echo "Autoboot set; going to multi-user. (DEL to exit.)"
          sleep 5
          init 2
fi

```

5. Save the file and exit the editor.
6. If you are not in the Pacific time zone, edit the time zone information in root's **.profile** file. The format of the time zone information is:

**TZ=xxxnzzz**

*xxx* is standard local time zone information, *n* is the number of hours the standard local time zone is west of Greenwich Mean Time (GMT), and *zzz* is the abbreviation of the daylight-saving time zone, if any.

The following table lists the time zones.

**TZ=PST8PDT** Pacific Standard Time

**TZ=MST7MDT** Mountain Standard Time

**TZ=CST6CDT** Central Standard Time

**TZ=EST5EDT** Eastern Standard Time

**TZ=MET-1MST** Middle European Time

**TZ=GMT0BST** Greenwich Mean Time (British Summer Time)

For example, if you live in Chicago, change the time zone line in **.profile** to:

**TZ=CST6CDT**

---

If you live in the Mountain Time Zone and DO NOT USE daylight savings time, change the line in **.profile** to:

**TZ=MST7**

If you live in the Eastern Time Zone and DO NOT USE daylight savings time, change the line in **.profile** to:

**TZ=EST5**

And finally, if you live in France, Germany, or Switzerland, (Middle European Time), change the line in **/etc/TIMEZONE** to:

**TZ=MET-1MST**

7. Log out, then log in as root. The system reads your **/.profile** file.
8. Check your environment. Enter:

**env OR set**

The system returns your environment.

---

## System States

As delivered, Plexus Sys5 is configured to run different system states. Each state is defined by what parts of the operating system are active at the time. Knowing what state your system is important because certain conditions require changing system states. Table 5-1 describes different states and the parts of the system that are active in each state:

**Table 5-1.** System States

State	Description
S, s, or 1	Single-user mode (system console only).
2	Multi-user mode. System console, all terminals, and system daemons are operational.
0, 3-6	State 0 and states 3-6 are for the user to define, if desired. These states are not used by Plexus.

## Finding the Current State

During normal multi-user mode, the system runs in state 2. You can find out what state you are in by entering:

```
who -r
```

The **-r** flag indicates the current run level of the **init** process. A display of the system state returned by the system follows:

```
. run-level 2 Jul 23 11:32 2 0 S
```

**[ NOTE ]** When you first boot the system, you are in single-user mode. Entering **who -r** at this point returns only a prompt. However, after you have gone to another state and back to **init 1** (single-user mode), entering **who -r** returns run-level 1.

---

If you are the super-user, you can find out the system state by entering:

**sys initstate**

The system returns the system state.

See the *Sys5 UNIX User's Reference Manual* for additional information on the **who(1)** and **sys(1)** commands.

### **Changing From State 2 to State s**

Sys5 provides a program called **/etc/shutdown**. By executing the **shutdown** program, you can change from state 2 (multi-user) to state s (system console only). **/etc/shutdown** sends appropriate messages to active terminals, kills all the necessary processes in the right order, and flushes the buffers. To shut down the system, perform the procedure "Shutting Down the System" described in the chapter *Startup and Shutdown*.

### **Changing From State s to State 2**

To take the system from state s (system console only) to state 2 (multi-user mode), follow the procedure in the chapter *Startup and Shutdown*.

### **Additional System Modes**

Besides the system states defined previously, **init(1M)** also accepts the following options (see Table 5-2).



---

**Table 5-2. System Modes**

<b>Mode</b>	<b>Description</b>
Q or q	Re-reads the <b>/etc/inittab</b> file immediately.
a, b, or c	Tells <b>init</b> to process only those <b>/etc/inittab</b> file entries having the a, b, or c run-level set. These are pseudo-states, which may be defined to run certain commands, but which do not cause the current run-level to change. For additional information see <b>init(1M)</b> and <b>inittab(4)</b> .

To re-read the **/etc/inittab** file, enter:

**telinit Q** or **telinit q**

This action quickly reads any changes made to the **/etc/inittab** file. **telinit**, which is linked to **init**, is used to direct the actions of **init**. **telinit** takes a one-character argument and signals **init** via the kill signal call to perform the appropriate action (in the above, re-reading the **/etc/inittab** file).

To run processes associated with a enter:

**telinit a**

See **init(1m)** in the *Sys5 UNIX Administrator's Reference Manual* for additional information on **telinit**.

---

## Auto Boot

Auto boot provides the capability to power up automatically to state 2. The auto boot feature is available on systems running Sys5 Rev. 1.3 or later.

To enable the auto boot feature, set switch 3 on the switch pack along the front edge of the CPU board to ON. Follow this procedure.

1. Shut down the system by executing the `/etc/shutdown` program (see the chapter *Startup and Shutdown*). Power the system off.
2. Set switch 3 on the switch pack along the front edge of the CPU board to ON (the fourth switch from the right since the switches are numbered starting with 0). This is easily done by pushing the dipswitch to the rear position with a straightened paper clip.
3. Power up the system (see the chapter *Startup and Shutdown*). When the system is powered up, it will automatically go to state s (system console only).
4. The system displays the following message on the console on 68020-based systems. On 68000-based systems, the message will be slightly different.

Auto-boot set. Going multi-user. (DEL to exit.)

There is a five-second period during which you can press the DEL key. If pressed, the system remains in the single-user mode.

If you do not press the DEL key, the system continues to multi-user mode.

---

**[ NOTE ]** The options to this auto boot invocation of **fsck** are **-y** and **-b**. These options tell **fsck** to assume a yes answer to all questions. This insures that all file systems will be repaired and the system will boot itself as many times as necessary to get a clean **fsck** of the root file system. If the **-y** and **-b** options are not in the **/etc/bcheckrc** script, edit the file **/etc/bcheckrc**.

Following is a sample portion of the **/etc/bcheckrc** file. What you need to add is in **boldface** type.

```
#          @(#)Kbcheckrc.sh                                6.1

# ***** This file has those commands necessary to check the file
# system, date, and anything else that should be done before mounting
# the file systems.

#
# If autoboot is set, assume date is correct and just check file systems.
# The '-b' option to fsck will cause fsck to re-boot system if root device
# gets updated
#
if sys autoboot ; then
    fsck -y -b
    exit
fi

export TZ
TZ= cat /etc/TIMEZONE
trap "" 2

# ***** Check date
```

---

## Modifying `/etc/rc`

Depending on your system's configuration, you might need to edit the `/etc/rc` file. Examples of different system configurations are listed in the following sections. Use these examples to determine if `/etc/rc` needs to be modified.

### Multibus Systems With ACPs and/or ICPs

To download a system with only ICPs, only ACPs, or ICPs and ACPs, you must modify the `/etc/rc` file to reflect the download sequence.

ICPs are first downloaded, starting with ICP0 and incrementing upward by one. Then, ACPs (if any) are downloaded starting with one more than the last ICP number and incrementing upward by one.

The command line in `/etc/rc` to download an ICP is:

```
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/icX -a 4000
```

*X* is the ICP number, beginning with 0

The command line in `/etc/rc` to download an ACP is:

```
/etc/dnld -L -da x y -f /usr/lib/dnld/acp -a 2000
```

*x* is the beginning ACP number (starting with 0)

*y* is the number of ACPs in the system

Following are examples of different configurations. Follow these examples and modify `/etc/rc` accordingly.

---

System With 3 ICPs:

```
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000  
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic1 -a 4000  
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic2 -a 4000
```

System With 1 ICP and 1 ACP:

```
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000  
/etc/dnld -L -da 1 1 -f /usr/lib/dnld/acp -a 2000
```

System With 2 ICPs and 2 ACPs:

```
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000  
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic1 -a 4000  
/etc/dnld -L -da 2 2 -f /usr/lib/dnld/acp -a 2000
```

System With 3 ICPs and 1 ACP:

```
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000  
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic1 -a 4000  
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic2 -a 4000  
/etc/dnld -L -da 3 1 -f /usr/lib/dnld/acp -a 2000
```

System With 1 ICP and 3 ACPs:

```
/etc/dnld -d -f /usr/lib/dnld/icp -o /dev/ic0 -a 4000  
/etc/dnld -L -da 1 3 -f /usr/lib/dnld/acp -a 2000
```

System With 0 ICPs and 4 ACPs:

```
/etc/dnld -L -da 0 4 -f /usr/lib/dnld/acp -a 2000
```

---

## VMEbus-Based Systems With VCPs

Each VCP needs a kernel downloaded to it when the system is initialized. The command line to download a VCP kernel is in the file `/etc/rc`. To download a VCP, the command line is:

```
/etc/dataio -dv x y -f /usr/lib/dnld/vcp -a 2000
```

*x* is the starting VCP number (beginning with 0)  
*y* is the number of VCPs in the system

Figure 5-1 lists two examples of the command line to download the VCP kernels. Figure 5-2 lists an example of the `/etc/rc` file.

Two VCP System:

```
/etc/dataio -dv 0 2 -f /usr/lib/dnld/vcp -a 2000
```

Four VCP System:

```
/etc/dataio -dv 0 4 -f /usr/lib/dnld/vcp -a 2000
```

**Figure 5-1.** Examples Downloading VCPs

---

## Sample /etc/rc for a VMEbus System

Following is a sample /etc/rc file for a VMEbus-based system. The download command lines are shown in bold.

```
/etc/cron
echo cron started
# /bin/su - adm -c /usr/lib/acct/startup
# echo process accounting started
#/usr/lib/errdemon
#echo errdemon started
# /bin/su - sys -c "/usr/lib/sa/sadc /usr/adm/sa/sa date +%d "
if [ $TimesInCurrentState -eq 0 ]; then
#
#       All VCP's must be downloaded and configured during system bootup.
#       Comment:-dv <starting vcp #> <# of vcp's>
#/etc/dataio -dv ?? -f /usr/lib/dnld/vcp5 -a 2000
#       To downline load 1 VCP:
#/etc/dataio -dv 0 1 -f /usr/lib/dnld/vcp5 -a 2000
#       To downline load 3 VCP's:
#/etc/dataio -dv 0 3 -f /usr/lib/dnld/vcp5 -a 2000
#rm /etc/vconfig/log
#       To configure asynchronous terminal lines use:
#/etc/vconfig/vconfig -cl /etc/vconfig/ASYNC
#       To configure parallel printer lines use:
#/etc/vconfig/vconfig -cl /etc/vconfig/PARA
```

**Figure 5-2.** Sample VMEbus-based /etc/rc File

## Modifying Maximum File Size

To prevent large files from being created by mistake, Sys5 includes a mechanism to limit the maximum size of any file which a process can create or access. The system administrator can allow individual users or all users to create and access larger files. The following procedure allows the system administrator to change the maximum file size limit of processes on a system.

1. Log in as root and position yourself in /etc.

- 
2. Enter the following C program and name it **unlimit.c**.

```
long ulimit();
main (argc,argv,envp)
int argc;
char **argv;
char **envp;
/*10000000 is the number, in blocks, of the new file size
system limit; you can use another block size*/
{ ulimit(2,10000000);
  execve("/etc/getty",argv,envp);
}
```

3. Compile this program using **cc** and place the executable file in the directory **/etc**. The following is an appropriate command line:

```
cc -N -O unlimit.c -o /etc/unlimit
```

Check to see if **unlimit.c** compiled correctly and if the executable file is in the **/etc** directory and has acceptable access permissions.

4. Modify **/etc/inittab**. Follow this sub-procedure.
  - a. Create a backup copy of **/etc/inittab**:

```
cp /etc/inittab /etc/inittab.bu
```
  - b. Invoke a text editor such as **ex** or **vi** to edit **/etc/inittab**.
  - c. Replace **getty** with **unlimit** for the user terminals you wish to have the privilege of creating large files.
  - d. Write and quit this file.
5. Tell **init** to re-read the **/etc/inittab** file by entering:

```
telinit q
```

6. Have each user log out and log back in. This is necessary because the old limit is in effect until each affected user logs



---

out and logs back in to the system.

7. If you are unable to change the system file size limit successfully, restore the original configuration and call the Plexus Software Support Center after verifying that all steps were performed correctly.
  - a. Run `/etc/shutdown` to bring the system down to single-user mode.
  - b. Save the new copy of `/etc/inittab`.
  - c. Move your backup copy of `inittab` (`/etc/inittab.bu`) to `/etc/inittab`.
  - d. Bring the system back up to multi-user mode.

---

## Accounting

Sys5 provides an extensive set of accounting programs that keep detailed records of user activities. For example, the Sys5 accounting system:

- provides methods to collect per-process resource utilization data
- records connect sessions
- monitors disk utilization
- charges fees to specific logins

Sys5 also provides a program to generate a system activity report.

Accounting is disabled at Plexus. See the chapter *Accounting* in the *Sys5 UNIX Administrator Guide* to enable accounting and define holidays and prime time.

When the accounting programs are enabled, the system does data reduction on the accounting data every evening so that disk space lost during the day is partially recovered at night.

**[ WARNING ]** The accounting programs use large amounts of disk space. If there is a shortage of disk space, consider obtaining more disk space or leave accounting disabled.

### Procedure to Enable System Activity Reports

Perform the following steps to enable the system activity report (sar):

1. Edit the file `/usr/spool/cron/crontabs/adm`
2. Find the lines that contain the strings `/usr/lib/sa/sa*` (\* is a number or letter).
3. These lines each start with a five fields to identify the exact time the command in the last field is to be executed. Change the fourth character in each field from a zero (0) to a star

---

(\*). For example:

Change 0 4 \* 0 \*  
to 0 4 \* \* \*

The fourth character occupies the field for months of the year. This step changes the value from 0 (none — an invalid or never true choice) to \* (all — always true).

If there is a pound sign ( # ) at the beginning of the line remove it. The # character at the beginning of a line comments the line. Removing the # uncomments the line.

4. Save your changes and exit the file.
5. No action is necessary to initialize this change. The file `/usr/spool/cron/crontabs/adm` is updated periodically by `cron(1M)` for commands to execute as described in the *Sys5 UNIX Administrator's Reference Manual*.

## Assigning Holidays and Prime Time

The accounting programs provide usage data divided into prime time and non-prime time. Prime time, usually defined as working hours on working days (when your system should be the busiest), can be set to whatever times desired. To assign holidays and prime time see the *Sys5 UNIX Administrator Guide* manual. The section "Updating Holidays" in the chapter *Accounting* explains how to assign holidays and prime time. The *Sys5 UNIX Administrator Guide* also provides additional information on accounting procedures.

---

## Using Tape Drives

This section explains how to use the 1/4-inch cartridge tape and the 1/2-inch reel-to-reel tape drives. You might not have both drives on your system. Read the subsections that are pertinent to your system configuration.

### Using the Cartridge Tape Drive

The cartridge tape drive uses self-contained 1/4-inch cartridges which greatly simplify operation compared with other tape drives. There are no controls, switches, or indicators associated with the tape. The controls are handled by software commands. Table 5-3 indicates the tape device names.

**Table 5-3. Cartridge Tape Device Names**

<b>Device Name</b>	<b>Description</b>
<b>/dev/rpt/0m</b>	Rewind (all systems)
<b>/dev/rpt/0mn</b>	No rewind (all systems)
<b>/dev/rpt/0fm</b>	Bit flipping, rewind (P/95 only)
<b>/dev/rpt/0fmn</b>	Bit flipping, no rewind (P/95 only)
<b>/dev/rft/0m</b>	Caching, Rewind (Multibus only)
<b>/dev/rft/0mn</b>	Caching, No rewind (Multibus only)

Note that on VMEbus-based systems, a bit flipping device driver is available. This allows you to read tapes made on Multibus systems.

The caching cartridge tape driver, available on Multibus systems with the IMSP controller, uses a file system on disk to buffer the data before reading or writing to tape. Both the disk and the tape drive must be attached to the IMSP controller. The caching cartridge tape driver will not work with disks controlled by an EMSP. The entire file system is used as a buffer. **DO NOT USE THIS FILE SYSTEM FOR ANYTHING ELSE OR YOUR DATA WILL BE OVERWRITTEN WHEN THE DRIVER IS USED.**

To use the caching cartridge driver, use **/stand/dconfig** to define the file system to be used by **/dev/rft/0m**. See the section *Creating Additional File Systems* in the chapter *Disk Configuration* for additional information. The special files **/dev/rft/0m** and *Getting Started*

---

`/dev/rft/0mn` can be created with the `mknod` command. The major number is 25. For `/dev/rft/0m`, use the same minor number as the file system you wish to use for the buffer. For `/dev/rft/0mn`, use 128 plus the same minor number as the file system you wish to buffer.

Create nodes for the two rft devices if the nodes are not listed in the `/dev` directory. See the examples below.

**mknod /dev/rft/0m c 25 15**

Creates a raw device for `/dev/rft/0m` (rewind) on `/dev/dsk/0s15`.

**mknod /dev/rft/0mn c 25 143**

Creates a raw device for `/dev/rft/0mn` (no rewind) on `/dev/dsk/0s15`. Note that  $128 + 15$  (for `0s15`) = 143.

**mknod /dev/rft/0m c 25 8**

Creates a raw device for `/dev/rft/0m` (rewind) on `/dev/dsk/0s8`.

**mknod /dev/rft/0mn c 25 136**

Creates a raw device for `/dev/rft/0mn` (no rewind) on `/dev/dsk/0s8`. Note that  $128 + 8$  (for `0s8`) = 136.

With `/dev/rft/0m[n]`, reads and writes take place asynchronously, occurring when the buffer area is filled. Therefore, errors which occur might not be reported until the tape device is closed. Be careful not to attempt to write more to the tape cartridge than it can hold. Tape errors are described in `pt(7)` and disk errors are described in `pd(7)` of the *Sys5 UNIX Administrator's Reference Manual*. For additional information on `/dev/rft/0m` and `/dev/rft/0mn` see `ft(7)` in the *Sys5 UNIX Administrator's Reference Manual*.

---

## Basic Operation

The tape cartridge itself has a switch labeled SAFE. If you turn the tab so the arrow points to SAFE, the tape can be read but not written to. Plexus recommends you set all software release tapes to SAFE to prevent losing your ultimate backup copies of system and application software.

Other than the safe switch, operating the cartridge drive consists of inserting the cartridge into the tape slot provided on the front panel of your system. Insert the cartridge with the label facing up. The safe switch and center pinch roller must face away from you (towards the rear of the tape slot).

---

## Features

P/35s, P/55s, P/75s, P/90s, and P/95s have 60Mb tape drives with 9 tracks. These units are streaming tape devices. On Multibus systems, the tape unit does not appear to be streaming during operation. Because cartridge tape drives operate to such close tolerances, head alignment, tape tension, and clean capstan and heads are critical factors in successfully transferring tapes from one drive unit to another. See your Maintenance Manual for tape drive maintenance information. Table 5-4 summarizes the cartridge tape drive information.

**Table 5-4. Cartridge Tape Drive Features**

Features	P/35	/55,P/75,P/95	P/90
Density	8000 bpi	8000 bpi	8000 bpi
Tape Speed	90 ips	90 ips	90 ips
Capacity	45Mb	60Mb	60Mb
Controller Manufacturer	Archive	Archive	NCR
Controller Interface	QIC-02	QIC-02	SCSI

## Retensioning

The tape drive manufacturer recommends that you retension the cartridge tapes at the following times:

- before first use
- before any read or write operation involving 2Mb or more of tape
- after two weeks of inactive storage

To retension the tape, insert the tape into the drive and enter:

```
/usr/plx/tape -f /dev/rpt/Xm retension  
X is the drive number
```

After the tape has been retensioned, the system returns:

```
DONE
```

---

## Using the Reel-To-Reel Tape Drives

This section describes how to use reel-to-reel tape drives. These tape drives are available in the following systems:

- P/60 system and expansion cabinet
- P/75 expansion cabinet
- P/95 expansion cabinet

Table 5-5 lists tape drives available.

**Table 5-5.** Reel-to-Reel Tape Drives

<b>System</b>	<b>Reel-to-Reel Tape Drive</b>
P/60	Cipher F880 (1600 BPI) Cipher M891 (1600, 3200 BPI)
<b>Expansion Cabinet</b>	<b>Reel-to-Reel Tape Drive</b>
P/60, P/75, P/95	Cipher M990 (1600, 3200, 6250 BPI)
P/75, P/95	Cipher M891 (1600, 3200 BPI)

Table 5-6 indicates the tape device names.

**Table 5-6.** Reel-To-Reel Tape Device Names

<b>Device Name</b>	<b>Description</b>
<code>/dev/rrm/0m</code>	reel-to-reel tape, rewind (all systems)
<code>/dev/rrm/0mn</code>	reel-to-reel tape, no rewind (all systems)
<code>/dev/rrm/0bm</code>	reel-to-reel tape, rewind, byte-swapping
<code>/dev/rrm/0bmn</code>	reel-to-reel tape, no rewind, byte-swapping
<code>/dev/crm/0m</code>	reel-to-reel caching tape, (rewind) (Multibus only)
<code>/dev/crm/0mn</code>	reel-to-reel caching tape, (no rewind) (Multibus only)

The `/dev/rrm/0bm[n]` drivers allow you to read tapes made on systems with different byte-ordering schemes.

To use the `crm` device, no set up is required. System main memory is used to buffer the data before reading or writing to tape. It is recommended to use the `crm` driver only for the stock P/60 tape drives.

Any combination of the following tape controls, as illustrated in Table 5-7, are supported on reel-to-reel tape drives.



---

**Table 5-7. Tape Controls Used With Reel-to-Reel Tape**

Bit #	Value	Meaning	Function
0	1	Drive Select	Selects drives 0 — 3
1	2	Drive Select	Selects drives 0 — 3
2	4	No Rewind	Does not rewind the tape when closing the device.
3	8	High Speed	Allows high speed movement of tape (not supported).
4	16	High Density	High tape density when density bits are set (not supported).
5	32	Extended Gap	Extends inter-record gap space. Use extended gap when writing to a newer and faster tape drive if the tape will be read from an older tape drive.
6	64	Not used	
7	128	Byte Swapping	A hardware supported option that swaps the high and low order bytes of each two byte pair. This allows you to read tapes made on systems with different byte-ordering schemes.

---

---

To illustrate how to make devices with **mknod**, a few examples are presented. To create the device driver for a high-speed, high-density, byte-swapped, extended gap, no-rewind device on drive 1, bits 0, 2, 3, 4, 5, and 7 are set on (10111101). Note that bit 0 is the right most bit. To create the following device, invoke the following **mknod** command.

```
mknod /dev/rrm/1hdgbmn c 8 189
```

**1hdgbmn** is the last part of the device name.  
You can use any name here.

**c** stands for character (raw) device.

**8** is the major device (character) for reel-to-reel tape.

**189** (minor dev #) = 1 + 4 + 8 + 16 + 32 + 128

To create the device drive for a high-speed, high density device on drive 2, bits 1, 3 and 4 are set on (00011010). To create this device, invoke the following **mknod** command:

```
mknod /dev/rrm/2hdm c 8 26
```

**2hdm** is the last part of the device name.  
You can use any name here.

**c** stands for character (raw) device.

**8** is the major device (character) for reel-to-reel tape.

**26** (minor device number) = 2 + 8 + 16

---

## Tape Drive Controls

A description of the tape drive controls follows.

**LOAD/REWIND** - Press to load or rewind the tape. The indicator flashes while the tape loads and remains lighted when loading is complete.

**UNLOAD** - Press to remove tape from the tape path. The indicator flashes as the tape clears the path and remains lighted when the tape path is clear.

**ON-LINE** - Press to place the tape drive under system control. The ON-LINE indicator will light. Pressing this switch a second time places the tape drive in the off-line mode, switching off the ON-LINE indicator. While on-line, the tape drive will not respond to other front panel controls.

**WRT EN TEST** - When this indicator is lit, a tape can be written to or erased. The drive senses the presence of a write-ring on the underside of the tape reel.

**DENSITY SELECT** or **HI-DEN** - Press to select the operating density of the tape drive. Available densities are: 1600 bpi (all drives), 3200 bpi (Cipher M990 and M891), and 6250 bpi (Cipher M990). The selected density is displayed on the front panel display.

**ADRS SELECT** - (Cipher M990 only) Press this switch to determine the tape drive address, from 0 through 7. This function is used only when the system is re-configured.

**Alphanumeric Display** - (Cipher M990 only) This is an 8-character display that provides information during normal operation and diagnostic testing.

---

**[ NOTE ]** The tape drive top cover and front-panel door lock when the tape drive is functioning. Any attempt to force either open before the tape is unloaded can cause mechanical damage to the loading system.

On some tape drives, the power switch is labeled 0 and 1 instead of OFF and ON.

### **Tape Drive Modes**

There are three normal modes of operation, each identified by the pattern of lights on the front panel. The modes and the light patterns visible during each mode are:

Idle/unloaded — Tape is unloaded, tape drive is idle and waiting for use. The UNLOAD light is ON; all others OFF.

Loaded for read — Tape is loaded correctly and ready for read operations only. The LOAD/REWIND and ON-LINE lights are ON; all others OFF.

Loaded for read and write — Tape is loaded correctly and a write ring is installed; tape drive is ready for read and/or write operations. The LOAD/REWIND, ON-LINE and WRT EN-TEST (write enable) lights are ON; all others OFF.

### **Improving Tape Drive Performance**

The following actions improve tape drive performance and decrease the chance of tape damage. This procedure is particularly important with 7.5-inch reels.

1. Prepare tapes using a standard half-inch-tape preparation tool. This tool rounds the end of the tape leader, increasing the chances of a successful automatic tape load. The tool can be purchased for approximately \$60 from most magnetic tape distributors or from Plexus. If a tape preparation tool is unavailable, use scissors to shape the end of the tape. It should be round with no sharp edges.
2. When inserting a tape reel into the drive, assure that the leader is fully wound onto the reel and not hanging over the

---

reel flange. If a wrap is loose from the reel before the air flow fans start operating, the tape may feed incorrectly and jam.

3. Use your index finger to hold the leader to the reel while inserting the reel. Once the tape is inserted and the finger removed, the tape can fall across the flange.

### **Automatic Loading**

During normal operation, the tape can be loaded automatically as follows:

1. Push the ON/OFF power switch at the front of the tape drive to the ON position. The system power ON/OFF switch must read ON (1) for power.
2. A self-test occurs whenever the tape drive is powered on. All indicators on the front panel of the tape drive light for several seconds. If no defect is found, all indicators except UNLOAD go OFF (unlit).
3. Wait for the UNLOAD indicator to light.
4. Wind the tape to be used completely onto the reel.
5. Gently flip down the top (center) of the front-panel door to open it.
6. Insert the tape into the tape drive with the write ring down (if there is a write ring on the tape).
7. Close the front-panel door.
8. Press the LOAD/REWIND switch; this locks the access doors. The LOAD/REWIND lamp blinks. After a delay, the WRT EN-TEST lamp lights (if a write ring is on the tape) and the LOAD/REWIND lamp stops blinking and stays ON. This indicates that automatic loading was successful, the tape BOT tab is at the photosensor and the self-test passed.

---

**[ NOTE ]** If the LOAD/REWIND lamp continues blinking for more than a minute, automatic loading has failed. Turn the tape power switch OFF, then back ON and repeat the loading sequence. If it fails three times, manual loading may be required.

9. When the LOAD lamp stops blinking and is ON steadily, press the ON-LINE switch to ON. The ON-LINE lamp will light. The LOAD/REWIND and ON-LINE indicators should be ON. If writing is desired the WRT EN-TEST indicator should also be ON. All other indicators should be OFF.
10. You can now read or write from your tape with commands like **cpio**, **finc**, or **dump**.

### Automatic Unloading

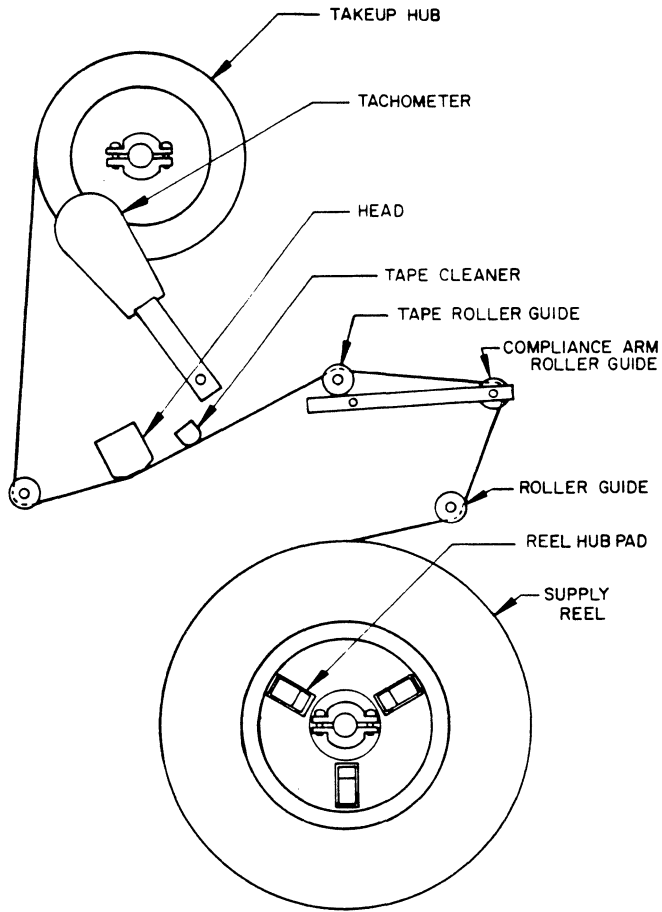
During normal operation, the tape drive unloads the tape automatically in response to the controls on the front panel. After copying files or file systems to tape, to unload a tape automatically, follow the procedure described below:

1. Ensure that the tape drive is off-line by pressing the ON-LINE switch until the ON-LINE indicator is OFF.
2. Press the UNLOAD switch. This switch unloads the tape from any point.
3. During the unload sequence, the UNLOAD indicator blinks and the access doors are locked. When the unload sequence is complete, the UNLOAD indicator remains lit and the access doors unlock.
4. Open the tape access door and carefully remove the tape reel.
5. Close the access door.

### Manual Loading

If the automatic loading procedure does not work, manual loading might be required. To load a tape manually:

- 
1. Place the drive in the maintenance-access position.
    - a. Push in the slide release latch located under the left lip of the tape drive front panel.
    - b. Withdraw the drive on its slides until the slide lock engages.
  2. Raise the top cover and secure it by placing the cover stay (on the left side) into its retaining slot.
  3. Place the tape reel on the supply side hub, making sure that the reel sits evenly on the hub.
  4. Turn the tape reel and hub clockwise while holding the hub lock button (located through the front door under the tape reel) until the reel is locked on the hub.
  5. Turn the tape drive ON/OFF switch to ON.
  6. Thread the tape (see accompanying diagram).
    - a. Carefully move the tachometer assembly away from the take-up hub.
    - b. Wrap several turns of tape clockwise around the take-up hub.
    - c. Gently replace the tachometer assembly on the take-up hub.
    - d. Check for correct seating of tape on guides.
    - e. Check for correct threading of tape through head assembly.
  7. Close top cover.
  8. Place transport in normal operating position by reversing the first two steps.
  9. Hold down the HI DEN switch (DENSITY SELECT on GCR Cache Tape Drives) while pressing the LOAD switch.
  10. The tape should tighten and advance until the BOT tab sits at the photosensor. The LOAD indicator lights to show that the transport is ready.



**Figure 5-3. Tape Unit, Tape Path**



---

## Manual Unloading

If the tape drive fails (usually due to a power failure or blown fuse), the tape must be removed manually. To remove a tape manually:

1. Power down the tape unit.
2. Place the tape unit in the maintenance-access position.
3. Open the front reel access door.
4. Rotate the tape reel counter-clockwise until all tape is rewound onto the reel.

**[ NOTE ]** If the tape reel will not rotate, depress the hub-lock release button (white button below tape reel).

5. Hold in the hub lock release button (see above NOTE) and rotate the tape reel clockwise until the three reel fingers move to a retracted position.
6. Remove the tape reel.

## Reel-to-Reel Tape Drive Error Messages

The following pages describe the reel-to-reel tape drive error messages during the load/unload procedure.

---

## Cipher M990 Error Messages

The Cipher M990 reel-to-reel tape drive is the standard drive housed in the P/60 expansion cabinet. Also, this drive is one of two possible reel-to-reel tape drives available in the P/75 expansion cabinet. The following error messages warn you of a problem.

**HUB SEAT** - Check to see whether the tape reel is properly positioned on the supply reel. After correcting the reel/supply hub alignment, press the Load/Rewind and Test switches to reset the error.

**LOAD FAILURE** - tape drive failed to load the tape. Check the tape leader for damage. Correct the damaged tape using the proper tool. Press the LOAD/REWIND, UNLOAD and TEST switches to reset the error.

**TAPE STUCK** - tape drive failed to unwind tape from the supply reel into the tape path. Check the tape leader. Press the LOAD/REWIND, UNLOAD and TEST switches to reset the error.

**NO DOOR LOCK** - indicates that the access door is open. Close the access door and the top cover before attempting to load the tape. Press the LOAD/REWIND, UNLOAD and TEST switches to reset the error.

**REEL UPSIDE DOWN** - tape reel inserted upside down. Remove the tape and insert with the write-ring facing downward. Press the LOAD/REWIND, UNLOAD and TEST switches to reset the error.

---

**NO BOT** - tape drive could no locate the load-point marker within the first 35 feet of tape. Make sure that the marker is within the first 6 feet of tape. Press the LOAD/REWIND, UNLOAD and TEST switches to reset the error.

**TACH ERR** - did not unpack packing contents from tape drive. The drive still has the packing material in it.

## Cipher F880 Error Messages

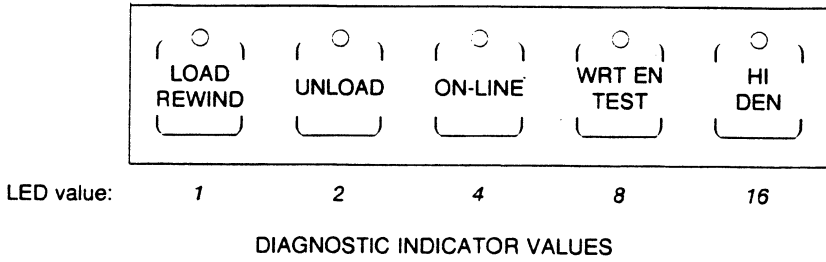
The Cipher F880 reel-to-reel tape drive is the standard tape drive housed in the P/60 main cabinet. With this tape drive and the Cipher M891 tape drive, abnormal operations are indicated by front panel display patterns. These error codes can be translated to binary codes based on the binary weight of the indicators (see the following figure). The following table lists error conditions during tape load.

**Table 5-8.** Cipher F880 Error Conditions During Tape Load

ERROR CONDITIONS DURING TAPE LOAD ( <i>Even on/off flash</i> )		
BINARY CODE	INDICATION	CONDITIONS
31	All indicators flashing	After four attempts, the drive did not successfully complete the load sequence. The tape leader should be checked for excessive damage. If a second attempt at autoloading fails, manually load the tape.
30	All indicators except LOAD flashing	The BOT marker was not detected within the first 35 feet of tape. The leader must be minimum of 6 feet in length.
29	All indicators except UNLOAD flashing	Tape reel was inserted upside-down. The bottom of the tape reel is indicated by the presence of an insertable write ring near the inside mounting radius.
27	All indicators except ON-LINE flashing	A load or unload operation was attempted with the front-panel door or top cover in the open position.
23	All indicators except TEST flashing	A load operation was attempted without inserting a tape reel into the transport.

---

The error codes can be translated to binary codes based on the binary weight of the indicators (see Figure 5-4).



**Figure 5-4.** Cipher F880 and M891 Diagnostic Indicator Values

---

### **Cipher M891 Error Messages**

The Cipher M891 reel-to-reel tape drive is one of two reel-to-reel tapes drives available in the P/75 and the P/95 expansion cabinet. Error conditions are listed in the following tables. See Figure 5-4 for diagnostic indicator values.

**Table 5-9. Cipher M891 Error Messages**

Cachetape Error Reference			
Error	Display	Description	Action
3	11000	The drive detected more than 3700 feet of tape	1. Try a different tape
4	00100	Compliance arm circuit voltage level is out of tolerance during the autoloading sequence.	1. Ensure that tape is properly wrapped around take-up hub. 2. Check compliance arm operation with Test 24.
5	10100	Sequence error: Read forward internal status self-check fault	
6	01100	The drive received a write or erase command for a file-protected tape	1. Possible host system failure. 2. Check if the write-enable ring is removed from the tape reel. 3. Check the file-protect circuit with Test 31 4. Check the interface logic with Test 21.*
7	11100	An illegal or undefined command was received by the drive	1. Possible controller or cable failure. 2. Check interface logic for floating or grounded inputs.*
8	00010	Unexpected done status in structure; internal status self-check fault	1. Note host command sequence, operating system, program version, release level, etc. 2. Contact factory
10	01010	Write Edit failure	1. New block size was greater than the original. 2. Re-check block size
11	11010	The number of write retries exceeded 16	1. Try a different tape. 2. Check write circuits using Tests 12 (tape unloaded) or 23 (tape loaded). 3. Check read circuits using Test 23 (tape loaded).*

**Table 5-9. Cipher M891 Error Messages (cont.)**

Cachetape Error Reference (cont.)			
Error	Display	Description	Action
13	10110	Illegal status found in structure. Internal status self-check fault	
14	01110	Tape travel past the EOT marker exceeded 18 feet	<ol style="list-style-type: none"> <li>1. Possible Host system failure.</li> <li>2. Check interface logic with Test 21.*</li> <li>3. Check EOT/BOT circuit using Test 32</li> <li>4. Ensure that EOT marker on the tape is properly placed</li> </ol>
15	11110	Data block exceeded the maximum block size allowed (32 KB).	<ol style="list-style-type: none"> <li>1. Possible host system failure (during write)</li> <li>2. Check ILWD interface input logic for a floating condition.*</li> </ol>
16	00001	Sequence error, read reverse; internal status self-check fault.	
17	10001	The compliance arm exceeded its travel limits during normal operation	<ol style="list-style-type: none"> <li>1. If unit missed EOT or BOT and tape ran off reel, check EOT/BOT circuit with Test 32.</li> <li>2. Check compliance arm operation with Test 24.</li> <li>3. Check servo operation with Test 11.</li> </ol>
18	01001	Tape speed variation in excess of the ANSI maximum of $\pm 10\%$	<ol style="list-style-type: none"> <li>1. Check servo operation with Test 11.</li> <li>2. Check tachometer operation with Test 11.</li> </ol>



**Table 5-9. Cipher M891 Error Messages (cont.)**

Cachetape Error Reference (cont.)			
Error	Display	Description	Action
19	11001	Vertical parity error on retries	1. Check write and read circuits using Test 23 (tape loaded).
20	00101	DMA failure. Word count not 0 after timeout.	1. Verify that Power up selftest (7) is completing successfully. 2. Check DMA and cache address/data lines using Test 42.*
21	10101	Excessive retries on write filemarks.	1. Readjust read threshold using Test 21 (tape loaded). 2. Check write formatter circuits with Test 13 (tape unloaded), not file protected, and writing 40 character blocks.
22	01101	Early EOT mark encountered (switch U3T-1 option).	1. EOT Marker located greater than 25 feet prior to actual EOT. 2. Relocate EOT marker.
23	11101	Hub not locked, too much slack tape during autoload.	1. Insert a tape reel into the drive. 2. Make sure tape reel is properly seated.
		Reel seat/file protect sensor failure during manual load.	1. Check if file protect or hub seat sensor are working properly.
24	00011	Parity error during Cache RAM refresh cycle. Generally, host cannot sustain throughput at present speed setting.	1. Check for "soft" RAM fault: cycle power to force power up check. 2. Check cache RAM with power up Selftest 7. 3. Rerun host program.
25	10011	Not enough tape on the takeup reel for manual load.	1. A minimum of five wraps is required.

**Table 5-9. Cipher M891 Error Messages (cont.)**

Cachetape Error Reference (cont.)			
Error	Display	Description	Action
26	01011	Tape stuck on the supply reel during autoload.	<ol style="list-style-type: none"> <li>1. Tape end did not peel off of reel. Remove antistatic tape/foam block if used.</li> <li>2. If caused by static charge buildup, refer to MANUAL load instructions.</li> </ol>
27	11011	Door interlock check.	<ol style="list-style-type: none"> <li>1. Close front panel door or top cover.</li> <li>2. Adjust door lock solenoid.</li> </ol>
28	00111	Servo failure or hub is jammed during manual load.	<ol style="list-style-type: none"> <li>1. Remove and re-insert tape reel to clear.</li> <li>2. Possible bell crank failure.</li> </ol>
29	10111	Reel upside down during load, or failure to get tape into path during autoload	<ol style="list-style-type: none"> <li>1. The bottom of the tape reel is identified by the write-enable ring groove or the write enable wring (when installed) near the inside mounting circumference.</li> </ol>
31	11111	After four attempts, the drive did not successfully complete the load sequence. Check tape end.	<ol style="list-style-type: none"> <li>1. The tape leader should be checked for excessive damage or static buildup.</li> <li>2. if a second attempt at autoloading fails, refer to MANUAL loading instructions.</li> </ol>

---

## Making Backups

Instructions describing how to use a variety of programs, including **cpio** to backup your files, are explained in the chapter *Backup Procedures*. Plexus Computers recommends that you use backup programs that recognize the end of file (EOF) mark. Although **dump**, **tar**, and **restor** are included on the release tape, these programs are not part of the standard release of Sys5 and were included for compatibility with Sys3. Plexus recommends that you use the Sys5 copy programs (**cpio**, **finc**, etc).

## SETTING UP I/O DEVICES

---

This chapter describes how to connect the following devices to your Plexus system:

- a terminal
- a parallel printer
- a serial printer
- an autodial (outgoing) modem
- a dial in modem

How to set up the system console is explained in the chapter *Getting Started*.

This chapter also:

- provides pinout tables needed to construct I/O cables
- explains how to configure VCPs (VMEbus-based systems only)

---

## Creating Nodes in /dev and Building /etc/inittab

On Multibus systems, to assist you with creating nodes in `/dev` and building your `/etc/inittab` file, Plexus created a utility, `/etc/ttyconf`, which is a shell script. `ttyconf` creates the appropriate devices in the `/dev` directory for the user-specified number of ICPs and ACPs.

On VMEbus-based systems, the program `/etc/vconfig/vconfig` creates nodes in `/dev` and configures the VCP. See the section "Configuring a VCP" in this chapter for additional information. Also see `vconfig(1M)` in the *Sys5 UNIX Administrator's Reference Manual*.

**[ WARNING ]** On Multibus systems, `/etc/ttyconf` should have been executed when your system was initialized. If you have not executed `/etc/ttyconf`, see your latest Sys5 release notice for further instructions. Do not invoke `mknod` to create nodes in `/dev`. `/etc/ttyconf` does this for you.

---

## Setting Up Terminals

To set up a terminal, do the following:

1. Ensure the terminal is set up properly. At the minimum, the following setup parameters should be established:
  - full duplex
  - eight data bits set
  - conversation mode enabled
  - no local echo
  - speed should match the speed in **/etc/gettydefs**
  - xon/xoff enabled
  - no parity
2. Set up a cable for your terminal. Consult the pinout tables in your terminal manual to determine what pins are required. See the pinout tables illustrated later in this chapter.
3. On Multibus systems, ensure that **/etc/ttyconf** has been executed. See the section "Creating Nodes in /dev and Building /etc/inittab". **/etc/ttyconf** creates nodes in **/dev** and builds the **/etc/inittab** file. On VMEbus-based systems, ensure that **/etc/vconfig/vconfig** has been executed. **/etc/vconfig/vconfig** creates nodes in **/dev**, builds the **/etc/inittab** file, and configures the VCP. See the section "Configuring a VCP" in this chapter for more information.
4. Edit **/etc/inittab**. Add a line to this file for the appropriate tty. Ensure that the action field is set to **respawn** and make sure the baud rate is set correctly. See the terminal manual for instructions on how to set the terminal to the correct baud rate. An example entry in **/etc/inittab** for tty 5 running at 9600 baud is:

```
04:2:respawn:/etc/getty /dev/tty5 9600
```

- 
5. Find or create a **terminfo** entry in **/usr/lib/terminfo/n/name**. Enter:

**cd /usr/lib/terminfo**

Change to the appropriate subdirectory (the first letter or number of your terminal type). For example, if you have a Wyse 100 terminal, change to the **w** directory. Then, list the contents of that directory. You will see **wy100**, which is the **terminfo** entry for the Wyse 100 terminal. If your terminal entry is not in **terminfo**, you must create (or have a programmer create) the entry. See the chapter *Curses and Terminfo Package* in the *Sys5 UNIX Programming Guide*. Also see **term(4)** and **terminfo(4)** in the *Sys5 UNIX Programmer Reference Manual*.

6. Plug in the terminal to the appropriate tty port.
7. If changes were made to the **/etc/inittab** file enter

**telinit q**

This action quickly reads the **/etc/inittab** file.

8. Turn the terminal on.

---

## ICP I/O Port Connections

This section describes the serial and parallel port connections.

### ICP Serial I/O Port Connections

All RS-232-C ports on the ICP are implemented using 25-pin DB-25F connectors, and contain all the signals required to support 3-lead tty or serial printer, asynchronous modems, and synchronous modems.

**Table 6-1.** ICP Serial Port Pinout

Pin #	Description	Direction
1	(not used)	
2	Transmitted data	TO ICP
3	Received data	FROM ICP
4	Request to send	TO ICP
5	Clear to send	FROM ICP
6	Data set ready	FROM ICP
7	Signal ground (common return)	(not applicable)
8	Carrier detect	TO ICP
15	Transmit clock	TO ICP
17	Receive clock	TO ICP
20	Data terminal ready	TO ICP

**Table 6-2.** ICP Terminal or Serial Printer Cable Pinouts

Computer DB-25	I/O Device DB-25	Signal Name
2	2	TXD
3	3	RXD
7	7	GND <sup>†</sup>
20	20	DTR <sup>†</sup>

<sup>†</sup> Use the DTR protocol for the serial printer cable only. The pinouts in Table 6-2 use the xon/xoff protocol.



---

**Table 6-3. ICP Asynchronous Modem Cable Pinouts**

<b>Computer DB-25</b>	<b>Modem DB-25</b>	<b>Signal Name</b>
2	3	RXD
3	2	TXD
4	5	CTS
5	4	RTS
6	20	DTR
7	7	GND
8	8	DCD
20	6	DSR

**Table 6-4. ICP Synchronous Modem Cable Pinouts**

<b>Computer DB-25</b>	<b>Modem DB-25</b>	<b>Signal Name</b>
2	3	RXD
3	2	TXD
4	5	CTS
5	4	RTS
6	20	DTR
7	7	GND
8	8	DCD
20	6	DSR
15	15	TXC
17	17	RXC

---

## ICP Parallel Connections

The parallel port uses a DB-37 connector. The ICP does not use Centronics standard connectors. Table 6-5 lists the ICP parallel port pinout connections.

**Table 6-5. ICP Parallel Printer Pinout Connections**

Parallel Connector Signals (Centronics Standard)			
Signal Name	Interface Pin#	Source	Description
DATA STROBE*	1,19	ICP	Pulse clocks data to printer
DATA 0	2,20	ICP	Data bit 0 <sup>†</sup> (LSB)
DATA 1	3,21	ICP	Data bit 1
DATA 2	4,22	ICP	Data bit 2
DATA 3	5,23	ICP	Data bit 3
DATA 4	6,24	ICP	Data bit 4
DATA 5	7,25	ICP	Data bit 5
DATA 6	8,26	ICP	Data bit 6
DATA 7	9,27	ICP	Data bit 7 (MSB)
ACKNLG*	10,28	Printer	Acknowledges receipt of character or end of function
BUSY	11,29	Printer	[NOT USED]
PE	12	Printer	Out of paper
SLCT	13	Printer	Printer selected
Gnd	14	Printer	Signal ground
Gnd	16	Printer	Signal ground
INPUT PRIME*	31,30	ICP	Initializes printer
FAULT*	32	Printer	Printer fault

NOTE 1. Second pin number (number after comma) indicates The ground wire of a twisted pair return.

NOTE 2. The symbol \* indicates *negative true* (i.e. logical NOT).

<sup>†</sup> low = 0, high = 1

---

## ACP I/O Port Connections

This section describes the ACP serial and parallel port connections.

### ACP Serial I/O Port Connections

The following tables list the pin connections required for fabricating serial I/O cables. The connectors required are male at the computer end (DB-9S) and usually male at the peripheral end (DB-25S). Check your device. Shielded cables are required to reduce electrical interference. Be certain that the cable shield is grounded at the peripheral end only.

The computer to terminal and serial printer cable described in Table 6-6 can be used with asynchronous serial terminals or serial printers on ACP serial I/O ports 0 through 15.

**Table 6-6. ACP Terminal/Serial Printer Cable Pinouts**

<b>Computer (DB-9)</b>	<b>To I/O Device (DB-25)</b>	<b>Signal Name</b>
1	3	RXD
6	2	TXD
8	7	GND

The computer to asynchronous modem cable described in Table 6-7 can be used with all ACP serial I/O ports 0 through 15.

---

**Table 6-7. ACP Asynchronous Modem Cable Pinouts**

<b>Computer (DB-9)</b>	<b>To Modem (DB-25)</b>	<b>Signal Name</b>
1	2	TXD
2	4	RTS
4	20	DTR
6	3	RXD
7	5	CTS
8	7	GND
9	8	DCD

The computer to synchronous modem cable described in Table 6-8 can be used with ACP serial I/O ports 0 through 7.

**Table 6-8. ACP Synchronous Modem Cable Pinouts**

<b>Computer (DB-9)</b>	<b>Modem (DB-25)</b>	<b>Signal Name</b>
1	2	TXD
2	4	RTS
3	17	RXC
4	20	DTR
5	15	TXC
6	3	RXD
7	5	CTS
8	7	GND
9	8	DCD

---

Table 6-9 lists the signal considerations when using the ACP board in a Plexus system:

**Table 6-9.** Data Carrier Detect and Clear to Send Signals

<b>Signal</b>	<b>Comments</b>
Data Carrier Detect	This signal is ignored for non-modem use and driven by the modem. For local ports on the ACP, edit the file <code>/etc/gettydefs</code> and specify <code>CLOCAL</code> in the second and third fields. For non-local ports (i.e. modems), ensure <code>CLOCAL</code> is <i>not</i> specified in <code>/etc/gettydefs</code> .
Clear To Send	This signal can be used to control character output (set with <code>stty</code> ) in asynchronous mode. The signal is used by many synchronous protocols to indicate when the system should output data to a synchronous communications modem.

---

---

## ACP Parallel Printer Port Connections

The parallel printer port provided by the ACP is a standard Centronics-parallel interface using a DB-37 connector.

**Table 6-10.** ACP Parallel Printer Pinout Connections

Parallel Connector Signals (Centronics Standard)			
Signal Name	Interface Pin#	Source	Description
DATA STROBE*	1,19	ACP	Pulse clocks data to printer
DATA 0	2,20	ACP	Data bit 0 <sup>†</sup> (LSB)
DATA 1	3,21	ACP	Data bit 1
DATA 2	4,22	ACP	Data bit 2
DATA 3	5,23	ACP	Data bit 3
DATA 4	6,24	ACP	Data bit 4
DATA 5	7,25	ACP	Data bit 5
DATA 6	8,26	ACP	Data bit 6
DATA 7	9,27	ACP	Data bit 7 (MSB)
ACKNLG*	10,28	Printer	Acknowledges receipt of character or end of function
BUSY	11,29	Printer	[NOT USED]
PE	12	Printer	Out of paper
SLCT	13	Printer	Printer selected
Gnd	14	Printer	Signal ground
Gnd	16	Printer	Signal ground
INPUT PRIME*	31,30	ACP	Initializes printer
FAULT*	32	Printer	Printer fault

NOTE 1. Second pin number (number after comma) indicates The ground wire of a twisted pair return.

NOTE 2. The symbol \* indicates *negative true* (i.e. logical NOT).

† low = 0, high = 1

---

## VCP/Link I/O Port Connections

Read this section if you have a VMEbus-based system (P/90 or P/95). This section:

- defines the serial cables you need to connect terminals, serial printers, and modems to your VMEbus-based system.
- describes the I/O port connections on the DB-25 parallel Link.
- describes how to configure the VCP, using the **/etc/vconfig/vconfig** program.

The following tables list the pin connections required for fabricating serial I/O cables. The connectors required are male at the computer end (DB-9S) and usually male at the peripheral end (DB-25S). Check your device. Shielded cables are required to reduce electrical interference. Be certain that the cable shield is grounded at the peripheral end only.

The computer to terminal and serial printer cable described in Table 6-11 can be used with asynchronous serial terminals or serial printers (DB-9 Link card [8 serial ports] and Telco Link card [16 serial ports]). Tables 6-11, 6-12, and 6-13 define the pinouts for the DB-9 Link card (8 serial ports).

---

**Table 6-11. Terminal/Serial Printer Cable Pinouts (DB-9)**

<b>DB-9 Link Card (DB-25)</b>	<b>To I/O Device (DB-25)</b>	<b>Signal Name</b>
1	3	RXD
6	2	TXD
8	7	GND

**Table 6-12. Synchronous Modem Cable Pinouts (DB-9)**

<b>DB-9 Link Card (DB-25)</b>	<b>Modem (DB-25)</b>	<b>Signal Name</b>
1	2	TXD
2	4	RTS
3	17	RXC
4	20	DTR
5	15	TXC
6	3	RXD
7	5	CTS
8	7	GND
9	8	DCD

**Table 6-13. Asynchronous Modem Cable Pinouts (DB-9)**

<b>DB-9 Link Card (DB-25)</b>	<b>Modem (DB-25)</b>	<b>Signal Name</b>
1	2	TXD
2	4	RTS
4	20	DTR
6	3	RXD
7	5	CTS
8	7	GND
9	8	DCD



---

## **P-4 Link Card I/O Port Connections**

To connect a parallel printer to your VMEbus Plexus system is extremely simple. Plexus has used the IBM standard female DB-25 connector on the P-4 end of the cable. The printer end of the cable is standard Centronics. You will need to purchase a standard DB-25 pin to Centronics cable. This cable, typically called an IBM PC<sup>†</sup> printer cable, is available at major computer dealers and electronic supply houses.

The cabling conforms to the convention established on the IBM PC for parallel ports. This also conforms to the RS-232 specification.

---

<sup>†</sup> IBM PC is a trademark of International Business Machines Corporation.

---

## Configuring a VCP

As part of the VCP's kernel, a system device table (SDT) contains all the necessary information to address a physical port. It is your responsibility to properly define each device name associated to a physical port. The information to define the ports is in the directory `/etc/vconfig`. Currently, only two configurations exist for a port: asynchronous and parallel. The file `/etc/vconfig/ASYNC` must be edited to reflect the configuration of your asynchronous serial ports. Likewise, the file `/etc/vconfig/PARA` must also be edited to define all of the parallel ports supported by your system's VCPs.

The first entry appearing in the file is the major device number. For the asynchronous port, the number is 31; for a parallel port, the number is 32.

The second entry in the file is the name of the protocol, a six character or less word. For an asynchronous port, the protocol is ASYNC; for a parallel port, the protocol is PARA.

The remaining lines are the I/O port configuration information lines. These lines are formatted as follows. Note that the field separator is a colon (:), space, or tab character.

```
I/O Port Name:VCP #:Link card #:Link Port #:Link ID
```

Link stands for Line interface card  
Link ID is described in Table 6-15

**Table 6-15.** Link ID Numbers

<b>ID #</b>	<b>Description</b>
DB-9	An eight port serial I/O Link card.
P-4	A four port parallel I/O Link card.

Figure 6-1 illustrates an example of the `/etc/vconfig/ASYNC` file. Characters appearing after a pound sign (#) are comments. You must edit these files to reflect your system's configuration.

---

```

31 #async major port number
ASYNC #protocol name
tty0:0:0:0:V8:# vcp 0, link card 0, port 0
tty1:0:0:1:V8:# port 1
tty2:0:0:2:V8:# port 2
tty3:0:0:3:V8:# port 3
tty4:0:0:4:V8:# port 4
tty5:0:0:5:V8:# port 5
tty6:0:0:6:V8:# port 6
tty7:0:0:7:V8:# port 7
tty8:0:1:0:V8:# vcp 0, link card 1, port 0
tty9:0:1:1:V8:# port 1
tty10:0:1:2:V8:# port 2
tty11:0:1:3:V8:# port 3
tty12:0:1:4:V8:# port 4
tty13:0:1:5:V8:# port 5
tty14:0:1:6:V8:# port 6
tty15:0:1:7:V8:# port 7

```

**Figure 6-1. ASYNC: One VCP and Two Link Cards**

Figure 6-2 displays an example of the `/etc/vconfig/PARA` file.

```

32
PARA
pp0:0:0:0:P4:# vcp 0, link card 0, port 0
pp1:0:0:1:P4:# port 1
pp2:0:0:2:P4:# port 2
pp3:0:0:3:P4:# port 3
pp4:1:0:0:P4:# vcp 1, link card 0, port 0
pp5:1:0:1:P4:# port 1
pp6:1:0:2:P4:# port 2
pp7:1:0:3:P4:# port 3

```

**Figure 6-2. PARA: Two VCPs with One Link Card Each**

---

## Loading an `/etc/vconfig` File

To fill the system device table (SDT), use the `vconfig` program. To load the file `/etc/vconfig/ASYNC`, enter the command:

```
/etc/vconfig/vconfig -cl /etc/vconfig/ASYNC
```

To load the file `/etc/vconfig/PARA`, enter the command:

```
/etc/vconfig/vconfig -cl /etc/vconfig/PARA
```

Plexus systems are shipped with the above commands in the `/etc/rc` file to automatically load the SDT during the boot-up process. Note that these commands are after the VCP download commands.

---

## Connecting a Parallel Printer

This section describes in a step-by-step manner how to connect a parallel printer to your Plexus system. It also explains how to set up the spooler. For additional information on the spooler, see the chapter *LP Spooling* in the *Sys5 UNIX Administrator Guide*. Follow these steps:

1. Make sure you have a parallel cable before you begin. Parallel cables can be ordered from Plexus Computers.
2. On Multibus systems, ensure you have run `/etc/ttyconf`. See the section "Creating Nodes in `/dev` and Building `/etc/inittab`" in this chapter. On VMEbus-based systems, ensure that `/etc/vconfig/vconfig` has been executed. `/etc/vconfig/vconfig` creates nodes in `/dev`, builds the `/etc/inittab` file, and configures the VCP. See the section "Configuring a VCP" in this chapter for more information.
3. Plug in the printer to the `pp0` connector using a parallel cable which can be ordered from Plexus. If you have an ICP-based system, see the CAUTION below.

**[ NOTE ]** If you unplug the parallel printer cable, after reattaching the parallel printer cable to the ICP you must do the following for the printer to function:

1. Shut down the system to single-user mode.
2. Reset the system by pressing RESET.

See the section "Shutting Down the System" in the chapter *Startup and Shutdown* for more information.

4. If you want to set up the spooler, follow these steps. Note that with each step you can verify setup of the printer with the `lpstat` command by entering:

```
lpstat -t
```

- 
5. After executing **lpstat -t**, if the scheduler is running, stop the scheduler. Enter:

```
/usr/lib/lpshut
```

to be sure.

6. Define the printer. That is, give it a name and a port using the **lpadmin(1M)** command. For additional information on **lpadmin**, see the *Sys5 UNIX Administrator Reference Manual*. Enter:

```
/usr/lib/lpadmin -pprinter -v/dev/ppX [-eprint | -iintfce | -mmodel ] [-cclass]
```

Note that the vertical bars ( | ) mean or. That is, use **-eprint** or **-iintfce** or **-mmodel**.

*printer* is the name you will use to refer to the printer

**/dev/ppX** is the device name (X corresponds to the parallel port number)

**-eprint** is a previously defined printer. This option tells the LP spooler to interface with this printer the same way as it interfaces with a previously defined printer, *printername* (i.e. emulates *printername*).

**-iintfce** establishes a new interface program for the printer. *interface* is the path name of the new program. If you prefer to use your own script to interface with the printer, create it in **/usr/spool/lp/interface** and use this option when defining your printer.

**-mmodel** selects a model script for the printer. *model* is one of the model interface names supplied with the LP Spooling Utilities: (**/usr/spool/lp/model/modelname**). Use the script **dumb** if none of the others is appropriate.

*class* is the class (group) of printers you want the printer to belong to. This is optional. Use the class option if you have identical parallel printers

---

and you want files sent to one printer automatically forwarded to another when the first is busy. For additional information see the chapter *LP Spooling* in the *Sys5 UNIX Administrator Guide*.

For example, to set up a NEC printer named `nec` on the first parallel port (parallel port 0) using the `dumb` model script and the class, `necclass` enter:

```
/usr/lib/lpadmin -pnec -v/dev/pp0 -mdumb -cnecclass
```

7. Make the printer the default printer to the LP spooling system with the `lpadmin(1M)` command. You must execute the `lpadmin` command twice: once with the `-p` option (see step 6) and once with the `-d` option. Enter:

```
/usr/lib/lpadmin -dprinter  
printer is the name used to refer to the printer
```

8. Allow requests to the printer. Enter:

```
/usr/lib/accept printer  
printer is the name used to refer to the printer
```

9. Start the spooler. Enter:

```
/usr/lib/lpsched
```

10. Ensure that the printer is on.

11. Enable the printer using the `enable(1)` command. Enter:

```
enable printer  
printer is name used to refer to the printer
```

This action activates the printer, thus enabling it to print requests taken by `lp`.

12. Uncomment the following entries in `/etc/rc`. That is, remove the `#` from the beginning of each line. When you are finished, the entries should look like:

```
rm -f /usr/spool/lp/SCHEDLOCK  
/usr/lib/lpsched  
echo "LP scheduler started"
```

- 
13. If you are printing a job and want to cancel the job determine the status of the job. Enter:

**lpstat -t**

The system returns a list of job numbers such as:

nec-prt158 on nec  
nec-prt159

on nec means the job nec-prt158 is currently being printed on the printer named nec.

14. To cancel nec-prt158, enter **cancel** followed by the information returned by the **lpstat -t** command. Enter:

**cancel nec-prt158**

15. You can add more than one job to cancel on the command line. For example, to cancel nec-prt158 and nec-prt159 enter:

**cancel nec-prt158 nec-prt159**



---

## Connecting a Serial Printer

To connect a serial printer to your Plexus system, follow these steps:

1. Set up a cable for your serial printer. To do this, consult the pinout tables in your serial printer manual. See Table 6-3 and Table 6-7 for ICP and ACP pinouts, respectively. If you plan to use xon/xoff flow control (Plexus default) use transmit, receive, and signal ground. If you plan to use data terminal ready to control data output use transmit, DTR, and signal ground.
2. On Multibus systems, ensure that `/etc/ttyconf` has been executed. See the section "Creating Nodes in `/dev` and Building `/etc/inittab`". On VMEbus-based systems, ensure that `/etc/vconfig/vconfig` has been executed. `/etc/vconfig/vconfig` creates nodes in `/dev`, builds the `/etc/inittab` file, and configures the VCP. See the section "Configuring a VCP" in this chapter for more information.
3. Plug in the cable to `ttyXX`.
4. Turn off login functionality in `/etc/inittab`. This makes sure `init` will not initiate a `getty` process for the port. To do this, invoke `vi` or another text editor and edit the `/etc/inittab` file. If you are connecting the serial printer to `tty13`, you must have an entry for `tty13`. Following is an entry for `tty13`:

```
13:2:off:/etc/getty /dev/tty13 9600
```

where:

- |            |  |
|------------|--|
| <b>13</b>  | is the id field used to uniquely identify an entry.  |
| <b>2</b>   | defines the run-level in which the entry is to be processed; run-level 2 means multi-user (init state 2).  |
| <b>off</b> | is the action field. It tells <code>init</code> how to treat the process specified in the process field. In this case the field should be off to disable (ignore) this line. |

---

**/etc/getty** is a program to be run. When **off** is specified, it is not run.

**/dev/tty13** is a parameter to **getty**. This is the device name for **/etc/getty**. In this example, the parameter is ignored because **off** is specified.

**9600** is a parameter to **getty**. This is an index into **/etc/gettydefs** which identifies the baud rate and other line parameters for **/etc/getty**. In this example, the parameter is ignored because **off** is specified.

If you are connecting the serial printer to **tty5**, the line in **/etc/inittab** would look like:

```
05:2:off:/etc/getty /dev/tty5 9600
```

5. If a **getty** was active on this port (i.e. previously was **respawn** enter:

```
telinit q
```

to have the new modification recognized by **init** and the **getty** process terminated.

In the steps that follow, you can verify setup of the printer with the **lpstat** command by entering:

```
lpstat -t
```

6. Be sure **lpsched** is not running. Enter:

```
/usr/lib/lpshut
```

to be sure.

7. Define the printer. That is, give it a name and a port using the **lpadmin(1M)** command. For additional information on **lpadmin**, see the *Sys5 UNIX Administrator Reference Manual*. Enter:

```
/usr/lib/lpadmin -ppntr -v/dev/ttyXX [-eprint | -iint | -mmod ] [-cclass]
```

Note that the vertical bars ( | ) mean or. That is use **-eprint** or **-iintfce** or **-mmodel**.

- 
- printer* is the name you will use to refer to the printer
- /dev/ttyXX* is the device name (*XX* corresponds to the serial port number)
- eprint* is a previously defined printer. This option tells the LP Spooler to interface with this printer the same way as it interfaces with a previously defined printer, *printername* (i.e. emulates *printername*).
- iint* establishes a new interface program for the printer. *intfce* is the path name of the new program. If you prefer to use your own script to interface with the printer, create it in */usr/spool/lp/interface* and use this option when defining your printer.
- mmod* selects a model script for the printer. **model** is one of the model interface names supplied with the LP Spooling Utilities: (*/usr/spool/lp/model/modelname*). Use the script **dumb** if none of the others is appropriate.
- class* is the class (group) of printers you want the printer to belong to. This is optional. Use the class option if you have identical parallel printers and you want files sent to one printer automatically forwarded to another when the first is busy. For additional information see the chapter *LP Spooling* in the *Sys5 UNIX Administrator Guide*.

For example, to set up a NEC printer named *nec* on *tty 4* using the **dumb** model script and the class, *necclass* enter:

```
/usr/lib/lpadmin -pnec -v/dev/tty4 -mdumb -cnecclass
```

- To set port characteristics, define where **stty** is going to be executed. The port characteristics might need to be set differently from the default. **stty** can be executed from */etc/rc* or in the */usr/spool/lp/interface/printername* program. It is recommended to set port characteristics in the interface program.

---

Following is an example of setting the port characteristics using **stty** for **tty14** in the interface program. This example works with VCP, ACP, and ICP systems.

```
stty sane 1200 ixon ixoff -ixany opost tab3 <tty14
```

If you edit **/etc/rc** instead of the interface program, you need to add set port characteristics information in **/etc/rc**. With ICP systems you also need to start **openup**. In Sys5 devices are special files. By running **openup** in the background, an entry for the device (i.e. special file such as **/dev/tty14**) is placed in the system's memory. With ICP systems, starting **openup** is necessary because when you invoke the spooler, the device (i.e. the special file such as **/dev/tty14**) is in the system's memory.

Following is an example of setting the port characteristics (ICP system) using **stty** for **tty14** in **/etc/rc**.

```
/etc/openup -w /dev/tty14
```

```
stty sane 1200 ixon ixoff -ixany opost tab3 <tty14
```

Because VCP and ACP systems use another method, it is not necessary to invoke **openup** in **/etc/rc**. Following is an example of setting the port characteristics for VCP and ACP systems using **stty** for **tty14** in **/etc/rc**.

```
stty sane 1200 ixon ixoff -ixany opost tab3 <tty14
```

9. Make the printer the default printer to the LP spooling system with the **lpadmin(1M)** command. You must execute the **lpadmin** command twice: once with the **-p** option (see step 7) and once with the **-d** option. Enter:

```
/usr/lib/lpadmin -dprinter
```

*printer* is the name used to refer to the printer

10. Plexus ports, by default, use the **xon/xoff** protocol. If you have a choice, set your printer for **xon/xoff**. If you must use signals, you must jumper the ICP. See the Maintenance Manual for additional information regarding jumpering ICPs. VCPs and ACPs are totally software configurable.

- 
11. Allow line printer requests. Enter:

```
/usr/lib/accept printer
```

*printer* is the name used to refer to the printer

12. Start the spooler. Enter:

```
/usr/lib/lpsched
```

13. Ensure that the printer is on.

14. Enable the printer using the **enable(1)** command. Enter:

```
enable printer
```

*printer* is the name used to refer to the printer

This action activates the printer, thus enabling it to print requests taken by **lp**.

15. Uncomment the following entries in **/etc/rc**. That is, remove the **#** from the beginning of each line. When you are finished, the entries should look like:

```
rm -f /usr/spool/lp/SCHEDLOCK
```

```
/usr/lib/lpsched
```

```
echo "LP scheduler started"
```

16. Try printing out a file, such as **/etc/group**, to make sure the printer works correctly.

17. If you are printing a job and want to cancel the job determine the status of the job. Enter:

```
lpstat -t
```

The system returns a list of job numbers such as:

```
nec-prt158 on nec
```

```
nec-prt159
```

**on nec** means the job **nec-prt158** is currently being printed on the printer named **nec**.

18. To cancel **nec-prt158**, enter **cancel** followed by the information returned by the **lpstat -t** command. Enter:

---

**cancel nec-prt158**

19. You can add more than one job to cancel on the command line. For example, to cancel nec-prt158 and nec-prt159 enter:

**cancel nec-prt158 nec-prt159**

---

## Setting Up Modems

This section describes how to set up autodial (outgoing) and dial in modems.

### Setting Up an Autodial Modem

The following procedure describes how to set up an autodial modem.

1. Ensure that the modem is strapped and auto answer is disabled.
2. Set up a cable for your modem. To do this, consult the pinout tables in your modem manual to decide what pins are required. Also see the pinout tables in this chapter.
3. On Multibus systems, ensure that `/etc/ttyconf` has been executed. See the section "Creating Nodes in `/dev` and Building `/etc/inittab`". `/etc/ttyconf` creates nodes in `/dev` and builds the `/etc/inittab` file. On VMEbus-based systems, ensure that `/etc/vconfig/vconfig` has been executed. `/etc/vconfig/vconfig` creates nodes in `/dev`, builds the `/etc/inittab` file, and configures the VCP. See the section "Configuring a VCP" in this chapter for more information.
4. With VCPs and ACPs, edit `/etc/rc`. Towards the end of the `/etc/rc` file, uncomment (remove the # from the beginning of the line) a command line at the Configure Modem Ports section of the script. Replace `tty##` with the correct tty number. The command line for tty 22 is:

```
stty -clocal </dev/tty22
```

The control mode `-clocal` assumes a line with modem control. A command line must be specified for each modem port on each VCP or ACP. For additional information see `stty(1)` in the *Sys5 UNIX User's Reference Manual*.

5. Ensure that the action field (the third field from the left) in `/etc/inittab` is set to off. Next enter:

---

### telinit q

This action re-reads the `/etc/inittab` file.

6. If you will be using **uucp** or **cu**, edit the file `/usr/lib/uucp/L-devices`. Following is a partial listing of the L-devices file:

```
DIR tty11 0 9600
DIR tty15 0 9600
ACU tty21 /usr/plx/dial 1200
DIR tty21 0 1200
```

The format of the **L-devices** file is:

*type line call-device speed*

where:

- |                    |   |
|--------------------|---|
| <i>type</i>        | is either DIR (directly connected to another system — hard-wired) or ACU (automatic call unit for autodial modems).   |
| <i>line</i>        | is the device name for the port.  |
| <i>call-device</i> | contains the name of the dialing program, <code>/usr/plx/dial</code> , if the ACU type is specified. If DIR is specified, this field is ignored. Place a 0 (zero) in this field with DIR. |
| <i>speed</i>       | is the baud rate.   |

For additional information, see the chapter *UUCP* in the *Sys5 UNIX Administrator Guide*.

7. Plexus ports, by default, use the xon/xoff protocol. With ICPs only, if you want to use other RS232 signals for flow control, you must jumper the ICPs. See your Maintenance Manual for jumper information. VCPs and ACPs are totally software configurable. There are no jumpers to set on VCPs or ACPs.



---

## Setting Up a Dial In Modem

To set up a dial in modem (a modem that will answer calls and allow people to log in), follow these steps.

1. Ensure that the modem is properly configured and auto answer is enabled.
2. Set up a cable for your modem. The pinouts are listed in this chapter.
3. On Multibus systems, ensure that `/etc/ttyconf` has been executed. See the section "Creating Nodes in `/dev` and Building `/etc/inittab`". `/etc/ttyconf` creates nodes in `/dev` and builds the `/etc/inittab` file. On VMEbus-based systems, ensure that `/etc/vconfig/vconfig` has been executed. `/etc/vconfig/vconfig` creates nodes in `/dev`, builds the `/etc/inittab` file, and configures the VCP. See the section "Configuring a VCP" in this chapter for more information.
4. With ACPs and VCPs only, edit `/etc/rc`. Towards the end of the `/etc/rc` file, uncomment (remove the `#` from the beginning of the line) a command line at the Configure Modem Ports section of the script. Replace `tty##` with the correct tty number. The command line for tty 22 is:

```
stty -clocal </dev/tty22
```

The control mode `-clocal` assumes a line with modem control (enter the above command line only with ACPs and VCPs). For additional information see `stty(1)` in the *Sys5 UNIX User's Reference Manual*. The above command line must be specified for each modem port on each ACP or VCP.

5. Ensure that the action field in `/etc/inittab` (the third field from the left) is set to `respawn`. Verify that the line speed label (the last parameter to `getty`) is the same as the desired label in `/etc/gettydefs`. For example, if you want to set up a 9600 baud modem, it is suggested to set the last parameter to `getty` in `/etc/inittab` to `M9600`, thus using a modem specific line speed label. The first field in `/etc/gettydefs` should also match the line speed label in `/etc/inittab`, `M9600`. To set up a 1200 baud modem, use

---

M1200 in **/etc/inittab** and **/etc/gettydefs**.

Re-read the **/etc/inittab** file. Enter:

**telinit q**

6. It might be necessary to edit **/etc/gettydefs**. The **/etc/gettydefs** file is set up using # to serve as field separators. Following is a description of the fields in **/etc/gettydefs**.

label # initial flags # final flags # login prompt # next label

where:

label	last parameter to <b>getty</b> in the <b>/etc/inittab</b> entry. It is usually the line speed of the terminal (i.e. 9600, 1200 etc.) This field is an argument to <b>getty</b> in the inittab file.
initial flags	indicates how to set up the terminal line when the login request is first made.
final flags	specifies the terminal line conditions before control is passed to the next process in the login sequence.
login prompt	this entire field is printed as the login prompt.
next label	This entry tells <b>getty</b> which entry to use next if the line speed seems incorrect (i.e. if the user presses the <BREAK> key).

A sample **/etc/gettydefs** file is illustrated in Figure 6-3.

---

```
console# EXTB HUPCL CS8 OPOST ONLCR # EXTB SANE IXANY TAB3 #Console Login: #console

M1200# B1200 HUPCL CS8 # B1200 SANE HUPCL TAB3 #Modem login: # M300

M300# B300 HUPCL CS8 # B300 SANE HUPCL TAB3 #Modem login: #M1200

1200# B1200 HUPCL CLOCAL CS8 # B1200 SANE CLOCAL TAB3 #login: #300

300# B300 HUPCL CLOCAL CS8 # B300 SANE CLOCAL TAB3 #login: #1200

9600# B9600 HUPCL CLOCAL CS8 # B9600 SANE CLOCAL TAB3 #login: #9600

19200# EXTA HUPCL CLOCAL CS8 # EXTA SANE CLOCAL TAB3 #login: #19200

2400# B2400 CLOCAL # B2400 SANE CLOCAL TAB3 #login: #1200

4800# B4800 CLOCAL # B4800 SANE CLOCAL TAB3 #login: #1200
```

**Figure 6-3. Sample /etc/gettydefs File**

7. Plexus ports, by default, use the xon/xoff protocol. If `uucp` is installed, use signals to monitor flow control (see `gettydefs(4)`).
8. If you want to use other RS232 signals for flow control (instead of xon/xoff), you must jumper the ICP. See your Maintenance Manual for jumper information and be sure to use correct cabling. ACPs and VCPs are totally software configurable. That is, there are no jumpers to set on ACPs or VCPs.

# OVERVIEW OF SYS5 FILE SYSTEMS

---

This chapter explains:

- an overview of physical and logical disks.
- file system organization (boot block, superblock, i-list, data block).
- how to use file-oriented system commands such as **ln**, **cp**, and **mv**.
- how a disk is configured by Plexus.
- what file systems Plexus defines in **/dev**.
- why file systems must be mounted or unmounted and how to use the **mount** and **umount** commands.
- how to determine the amount of disk free space on your disk.
- how to use the **ramdisk** utility, which allocates system memory as a disk.

This chapter provides an overview of file systems. After reading this chapter, see the chapter *Changing Root and Swap* for step-by-step instructions describing how to change the location of the root file system and/or the swap area, and the chapter *Configuring Disks* for step-by-step instructions explaining how to set up additional file systems (for user accounts, application programs, etc.).

---

## Physical and Logical Disk Overview

In Sys5, files are organized on a partition of the disk called a file system. Each physical disk can be divided into multiple sub-disks called logical disks. File systems are set up on these logical disks.

Physical disks on a Plexus system are numbered starting with 0 and can go through number 15. The disk naming convention follows:

`/dev/dsk/nsnn`

Where: `/dev` refers to the `/dev` directory  
`/dsk` is the special file name for block I/O disk drives  
`n` is the physical disk number  
`nn` is the logical disk (partition) number

Thus, for each physical disk up to 16 logical disks can be created. The first *literal* logical disk (`ns0`) is set up by Plexus to refer to the entire physical disk. Plexus never uses this first partition as a file system. By Plexus convention, the second literal logical disk, `ns1`, is the first logical disk on each physical disk on which you can set up a file system. The number of logical disks you create per physical disk will depend on the size of your disk. You would probably divide a 586Mb drive into more logical disks than you would a 71Mb drive.

The Plexus convention for naming physical disks follows.

- `/dev/dsk/0s0` references the entire physical disk. Plexus never uses the first partition, `ns0`, as a file system.
- `/dev/dsk/0s1` is thus considered the first logical disk on disk 0. Plexus uses `0s1` for the root file system.
- `/dev/dsk/0s2` is considered the second logical disk on disk 0. Plexus uses this for the swap space (Rev 1.3 or later).
- `/dev/dsk/0s3` is user definable. You can create a file system on `/dev/dsk/0s3` to store related files and directories. Subsequent logical disks can be set up in the same way. WHEN SETTING UP A FILE SYSTEM FOR USER ACCOUNTS DO NOT USE THE `/usr`

---

FILE SYSTEM FOR USER ACCOUNTS. Instead, use `/user`.

The numbering convention continues through:

`/dev/dsk/0s15` is user definable. 0s15 is the last logical disk on physical drive 0.

Each logical disk is treated as if it were a separate physical disk drive. These contiguous logical disks might or might not be structured into a file system so that data can be accessed by the user via an extension of the hierarchical directory structure.

Some application programs, such as UNIFY, support the use of raw disk to store database information its own way. In this case, the logical disk area is not formatted into a file system structure.

### Systems With IMSP and EMSP Controllers

On Plexus Multibus systems with both IMSP and EMSP controllers (P/60s and P/75s), the default controller is the EMSP. Thus, when you enter

```
/dev/dsk/Xsxx
```

the system accesses disks controlled by the EMSP disk controller. To access IMSP-controlled disks, instead of the generic device name `dsk`, use `pd`, which is the device name for the IMSP controller. Enter:

```
/dev/pd/Xsxx
```

If you have a system with 10 physical disks (two IMSP-controlled disks and 8 EMSP-controlled disks) access the IMSP-controlled disks as follows:

```
/dev/pd/0sxx
```

```
/dev/pd/1sxx
```

---

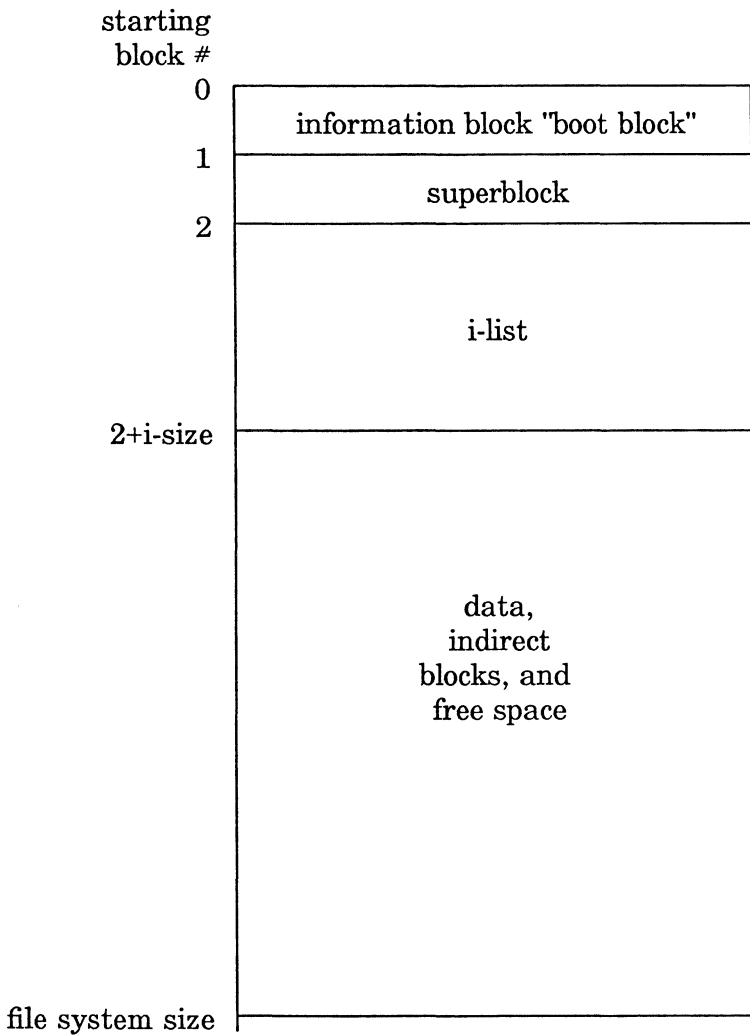
Access the EMSP-controlled disks as follows:

```
/dev/dsk/0sxx  
/dev/dsk/1sxx  
/dev/dsk/2sxx  
/dev/dsk/3sxx  
/dev/dsk/4sxx  
/dev/dsk/5sxx  
/dev/dsk/6sxx  
/dev/dsk/7sxx
```

Note that the disk numbering convention begins with disk 0 for both disk controllers (**/dev/pd/0sxx** and **/dev/dsk/0sxx**). For a complete list of disk device names, see the chapter *Stand-alone Programs*.

## File System Organization

File systems represent a structured organization of data. However, the layout is much different than appearance suggests. Information is actually stored as a contiguous array of blocks, each comprised of two 512-byte sectors and referenced as a single entity by a special block device node called **dsk**, which is contained in the **/dev** directory. The device file **/dev/dsk/0s1** references the root file system, or the part of the disk containing the Sys5 root files and data. Figure 7-1, following, sketches out the basic organization of the file system.



**Figure 7-1.** mkfs File System Organization

In the previous figure, *i-size* is the length of the i-list, and *file system size* is the length of the file system. More detailed information follows.



---

## Boot Block

The boot block is the first block of the file system. Although every file system has a boot block, only the boot block on the logical disk used to format the disk (Os1) contains valid information. The boot block on Os1 stores the necessary information about the disk as a whole (disk partitions and boundaries) and the location of the files necessary to bring the system up. Boot blocks on other file systems are never initialized or used. The contiguous blocks of information in a file system are grouped together into: the superblock, the i-list, and the data area.

## Superblock

The superblock is always the second block of the file system. The superblock contains information about the file system itself: file system size, the number of i-nodes, data blocks, and unused blocks of the file system.

## i-list

The i-list starts at block 2 and is a string of i-nodes. I-nodes are the principle file system references to a file. An i-node consists of a 64-byte description of the file. The i-node includes pointers (addresses) to where the data contained in the file is located. Information stored in the i-node includes:

- a unique i-node number
- the file size
- the file type (ordinary file, directory, or special file)
- the protection sequence
- the number of links (or the number of directories pointing to that file)
- the user ID number (UID)
- the group ID number (GID)

- 
- the date the file was last modified and accessed
  - address locations to the data area on the disk where the file's contents are stored.

Most of the information displayed by the list command is actually read from the i-nodes of the files within that directory. Directories are a type of file that give hierarchical appearance to the file system. A directory is simply a reference file containing the i-node numbers and names of the files contained within it.

## Data Area

The data area contains the blocks of data that are in the file itself, and empty blocks. This data (text, executable code, or other character formats) is stored in a block fashion. Each block represents two 512-character/byte segments or sectors. Note that the data itself is on a separate part of the disk from the file's i-node information. These data blocks are pointed to (or referenced by) the corresponding file's i-node. So, the file system knows where to find the information contained in the file when the user calls for it.

## Indirect Blocks

The data area of the disk also contains indirect blocks which allow indexing of pointers to data for files containing large amounts of information. The file's i-node contains 13 addresses (pointers) to the data area containing the information. The first 10 addresses point to the first 10 blocks of information. If the file is larger than 10240 bytes (1024 X 10), then the eleventh address is used to point to an indirect data block, which in turn, contains pointers to the next 256 blocks of the file's data. If it is even larger than this, then the twelfth address points to an indirect block which points to another double-indirect block, whereby the first indirect block contains up to 256 blocks of the file. The thirteenth address is reserved for triple indirection which is used for larger files. With this indirection in mind, it is easy to see why the file system concept allows the ease of access to small and large files.

---

## Free Space

At the end of the data area is the free space. Free space represents the unused data area of the disk and also serves as an indicator of how much more information can be stored in the file system. After the file system has been used a while, free space is not usually contiguous.

For each file system resident on the disk, there is a section with all the i-nodes strung together followed by a section with all the data strung together formatted in 1024 character units. Each i-node points to the proper data block(s) associated with the file's contents.

For additional information on file system organization, see the article on **fsck** in the *Sys5 UNIX Administrator's Guide*.

---

## File Systems — a Closer Look

With the concept of file systems in mind, we can accurately explain what some of the file-oriented Sys5 commands actually do. After an explanation of file naming conventions, this subsection explains three ways files can share information:

- linking files
- copying files
- moving files

### File Naming Conventions

A file name should contain 14 or less characters including the letters A — Z, a — z, the numbers 0 — 9, periods ( . ) and underscores ( \_ ). DO NOT USE any special character listed in the following note.

**[ NOTE ]** Do not use the following special characters in file names: asterisk ( \* ), question mark ( ? ), exclamation point ( ! ), left bracket ( [ ), right bracket ( ] ), blank space ( ), tab character, redirect from ( < ), redirect to ( > ), backslash, ( \ ), single quote ( ' ), double quote ( " ), accent grave ( ` ), vertical bar ( | ), caret ( ^ ), left curly brace ( { ), right curly brace ( } ), hyphen ( - ) parentheses ( ( ) ), comma ( , ), slash ( / ), pound sign ( # ), dollar sign ( \$ ), at sign ( @ ), and ampersand ( & ). Also, do not use non-printing characters, such as control characters (i.e. CTRL d), in filenames.

---

## Linking Files

Sys5 allows you to link files together. This means that two different file names actually reference the same data blocks. When you link an existing file to a new file with the command:

```
ln oldname newname
```

you are simply creating another pointer to an existing file. The **ln** utility does not make another copy of the file *oldname*. Because there is only one file, the file status information such as access permissions, ownership, and the date the file was last modified for *oldname* and *newname* is the same.

If you link two files in the same directory, the file names of the two files must be different. However, if you link two files in different directories, the file names can be the same. Linking files is much like creating an alias for the original file. To you, it might look as if a separate file was created in its entirety. But, this is not the case. Using the **i** option with **ls** (i.e., **ls -li** or **ls -i**) displays the i-node number which reveals the duplicate i-node directory entry.

For example, to link a file named **report**, which resides in **/user/jeff/report**, to **/user/joyce/draft**, enter:

```
ln /user/jeff/report /user/joyce/draft
```

The file **/user/jeff/report** is now linked to **/user/joyce/draft**. These two files are actually different references to the same data (i-node). To verify this enter:

```
ls -li /user/jeff/report /user/joyce/draft
```

The system responds:

```
1364  -rw-rw-rw-    2  jeff   100   463   Jun 5 07:29    /user/jeff/report
1364  -rw-rw-rw-    2  jeff   100   463   Jun 5 07:29    /user/joyce/draft
```

Note that the i-node number for both files, 1364, is identical.

Linking to a file gives you the existing permissions, and owner and group id of the file you link to. The original i-node information is maintained in the link. In the example above, Jeff owns the file but anyone can read from or write to the file because the

---

read/write permissions are set `rw-rw-rw-`. This is sharing common data with the reference to that file in two or more directories. After linking the file, if the read/write permissions do not allow you to read and write to the file, change the read/write permissions with **chmod**.

The number in the third column of an `ls -li` listing on a directory shows you the number of existing links to that file. If the number is 2 or greater, this means that someone else is using the data associated with that file description; in other words, they are linked. Directories always will have a minimum link count of 2. This is because the directory is linked to itself and its parent directory in order to have a valid place in the hierarchy. Each sub-directory increases the link count of the main directory containing it by one (because of the parent reference).

It is logically impossible to link files across file systems, due to the separate nature of file systems themselves. In other words, the file i-node number 234 is a much different file on the file system residing on `dev/dsk/0s1` than i-node number 234 on the file system residing on `/dev/dsk/0s3`. I-node numbers and data blocks are only known and accessible within the same file system.

## Copying Files

Copying a file with the command:

```
cp oldfile newfile
```

is not the same as linking a file. Invoking the **cp** command creates two distinct versions of the file. The **cp** command creates a new i-node and replicates the data blocks for the file copy. It should be fairly clear that copying a file uses up much more disk space than linking files simply because linked files share one common i-node and data space (i.e. contents are shared). Copied files are entirely separate with distinct i-nodes and separate data space allocation. They are only similar by what data is stored in the data space, until modification of either copy takes place. Therefore, it is possible to copy files from one file system to another.

---

To illustrate the **cp** command, a file named **demo** in the directory **/user/joyce** was copied to **/user/joyce/letters/demo1**. When you enter:

```
ls -li /user/joyce/demo /user/joyce/letters/demo
```

The system returns:

```
1084 -rw-rw-rw- 1 joyce 1146 Sep 23 10:02 /user/joyce/demo
1588 -rw-rw-rw- 1 joyce 1146 Sep 23 10:03 /user/joyce/letters/demo1
```

Note that the i-node numbers, 1084 and 1588, are different.

## Moving Files

When you move a file within the same file system with the command:

```
mv oldfile newfile
```

the i-nodes and data blocks remain the same; you are just changing the directory name reference to that file (hence changing its name, location, or both). When the **mv** command takes place, the file's i-node number itself remains untouched; the i-node-number-file-name reference in the directory is deleted and a new one created. Thus, the appearance is given that the file has been actually moved. When a file is moved to another file system, the file system behaves as if the file was copied to the other file system, then removed on the original one.

---

## Additional File Systems

Plexus systems are shipped from the factory with the root file system set up on `/dev/dsk/0s1` and the swap area on `/dev/dsk/0s2`. To use the rest of the disk (from `/dev/dsk/0s3` to `/dev/dsk/0s15`) you need to create additional file systems. Sys5 treats all file systems exactly as if they resided on separate disk drives (physical partitions). Since Plexus has no idea how many additional file systems you wish to configure, the task of making additional file systems is left up to you.

The following factors should enter into your decision of how many additional file systems to create:

- Accessibility
- Backup Schemes
- Protection
- Disk Capacity

Each of these factors will be discussed in the following subsections.

### Accessibility

Each file system containing files (i-nodes and data) is completely independent of the other file systems. As such, the files on your additional file system can be attached (mounted) to and detached (unmounted) from directories on your root file system. It is therefore possible to make all directories and files in these additional file systems accessible to the system users at the system administrator's discretion.



---

## Backup Schemes

Each file system must be backed up on a separate tape. Therefore, if you have a large number of files which do not get updated on a regular basis, you could load them into a separate file system (i.e. `/dev/dsk/0s4`) and back that file system up on a more infrequent basis than another file system containing your system user directories (i.e. `/dev/dsk/0s3`). That way, backups are much faster. A popular configuration is to have user directories (which tend to have rapid file turnover) in one file system and application programs (which rarely change after installation) in another file system.

## Protection

Important data is easier to recover (when lost by user error or disk failure) if your data is organized into file systems. For example, one of your file systems can be totally corrupted without affecting files on the other file systems. Since individual file system backup tapes are shorter, it is much faster to restore individual lost files or complete file systems. Some administrators create an additional file system simply for the purpose of backing up crucial data on another equivalent file system. This is referred to as disk-to-disk copy or hot backup.

## Disk Capacity

The storage capacity of your disk has much to do with the number of file systems you should create and how large they can be. If, for example, you have a 36Mb disk, you will probably only want to create one additional file system (`/dev/dsk/0s3`). A file system should be at least 10000 blocks in size. Extremely large file systems result in lengthy disk-to-tape backups and file checking/repairing procedures.

---

## Mounting File Systems

The root file system contains the system files that comprise the Sys5 operating system. When Sys5 is first brought up into single-user mode, only the root file is system accessible. The other file systems reside on different parts of the disk or on different disks. Until called upon, they are unknown by Sys5. In order to access a non-root file system, the device referencing it must be mounted onto a directory in the root file system.

In order to mount a device, an empty directory must be created for it. To access the contents of a file system referenced by the disk device `/dev/dsk/0s6`, use the `mkdir` command to create a "mountstub" directory which is solely for the purpose of attaching the `/dev/dsk/0s6` file system to the Sys5 directory hierarchy on `/dev/dsk/0s1`. Now use the `mount` command to mount the `/dev/dsk/0s6` file system to that directory. We will call this directory `/u`. At the root prompt (`#`), enter:

```
mkdir /u
/etc/mount /dev/dsk/0s6 /u
```

Any reference to `/u` now references directories and files on the file system `/dev/dsk/0s6`. List (`ls`) the contents of the `/u` directory and files stored on logical disk `/dev/dsk/0s6` will display.

The opposite of the `mount` command is `umount`. This command detaches, or unmounts, a file system. For example, enter:

```
/etc/umount /dev/dsk/0s6
```

and all of the files on the `/dev/dsk/0s6` file system become unavailable. When you list the `/u` directory, it will be empty. The `/dev/dsk/0s6` file system is detached from `/dev/dsk/0s1`.

File systems that you want mounted when you enter multi-user mode, and unmounted when you enter single-user mode, should have `mount` and `umount` commands set up in the `/etc/rc` file. This shell script is run when the system is taken from single-user to multi-user mode, and vice-versa. When the `mount` and `umount` commands are listed in the `/etc/rc` file, the file systems will automatically be mounted and unmounted at appropriate times. Figure 8-7 shows this portion of the `/etc/rc` file.

---

## Mounting Considerations

The root file system, `/dev/dsk/0s1`, mounts automatically when you boot into single-user mode. It cannot be unmounted as long as the system is booted. Alternate file systems (such as `/dev/dsk/0s3`) can be mounted or unmounted by the superuser in single-user or multi-user mode. Usually, the additional file systems are set up to automatically mount in multi-user mode and unmount in single-user mode.

Executing the **mount** command without any arguments will cause the system to display a list of the file systems that are mounted. Executing the **list** command (**ls**) on the `mountstub` directory will display nothing if the file system is not mounted; if files or directory names display, the file system probably is mounted. There should be no files or directories in `/u` while the file system is unmounted.

When you create a new file system (or when you restore an existing file system from tape), you must create a **lost+found** directory with sufficient room. **lost+found** must be created when the logical block device is mounted to the file system. The **lost+found** directory is used by the file system consistency check program, **fsck**, to deposit files that are somehow disconnected from their original directory. In other words, it provides a place where the **fsck** program can place orphaned files.

In a hierarchical sense, your new file system, when mounted, contains all of the files and directories in the `mountstub` directory (`/dev/dsk/0s6`). The root file system (`/dev/dsk/0s1`) contains every file and directory, including the empty `mountstub` directory (`/u`), but the contents of the `mount stub` directory—those files and directories in `/u`—will be stored in the new file system (`/dev/dsk/0s6`). However, if `/dev/dsk/0s6` is unmounted and you create files under the `/u` `mountstub` directory, they will be stored on `/dev/dsk/0s1`, not `/dev/dsk/0s6` (because `/u` is just another empty directory on the root file system). When `/dev/dsk/0s6` is mounted onto `/u`, the files on `/dev/dsk/0s1` will be overlaid with the files on `/dev/dsk/0s6`—the files on `/dev/dsk/0s1` will seemingly disappear until an unmount is performed. Therefore, be certain that `/dev/dsk/0s6` is mounted onto the `mountstub` directory

---

before creating any files or directories. While unmounted, the mountstub directory should be empty.

You can create more than one additional file system, configured and mounted simultaneously. The maximum number of file systems that can be created on a single disk is 16 (0 - 15); of those 16, only a certain number can be mounted at any one time. NMount in the file `/usr/src/uts/m68/cf/conf.c` defines the number of files that can be mounted at the same time. Each file system will have its own separate mountstub directory in `/dev/dsk/0s1`; for example, `/dev/dsk/0s6` could be mounted onto `/u`; `/dev/dsk/0s3` might be mounted onto `/user`, etc.

If you try to mount a file system that is already mounted, or while someone is in the mountstub directory, you will receive the following error message:

```
WARNING!! -mounting: < > as /u
mount: Device busy
```

If you see a WARNING message when mounting, it means that the mount was successful, but the file system has not been labeled to mount onto that particular mountstub directory. See the following chapter for information on how to label the new file system, or refer to the **volcopy(1M)** command in the *Sys5 UNIX Administrator's Reference Manual* for more information about mounting file systems.

If any of the files in a file system are open (being used), the **umount** command will not let you detach that file system from the mountstub. If you try to unmount the `/dev/dsk/0s6` file system while someone is it, you will receive the message:

```
umount: /dev/dsk/0s6 busy
```

If you try to unmount the file system `/dev/dsk/0s6` and it is not mounted, you will receive the message:

```
umount: /dev/dsk/0s6 not mounted
```

The message:

```
warning: umount: /dev/dsk/0s6 was not in mount table
```

---

indicates that although the unmount was successful, the file system was not registered/entered in the `/etc/mnttab` file. `/etc/mnttab`, a data file which cannot be read directly, keeps track of which file systems are mounted and where they are mounted. You can ignore this message because the mount table is re-initialized when you change from single-user to multi-user and vice-versa. You can use the command `setmnt(1M)` to do this yourself (see the *Sys5 UNIX Administrator's Reference Manual*. For additional information on `/etc/mnttab` and `setmnt(1M)`, see the next section "File System Status".

**[ NOTE ]** Do not attempt to edit the `/etc/mnttab` file directly. It is not a text file.

Reasons to unmount an additional file system are:

1. Perform any file system repair with the additional file systems unmounted and the system in single-user mode. This ensures that no files in the file system are active during the check/repair procedure.
2. Depending on how `/etc/rc` is set up, the additional file systems should be automatically unmounted when you **shut-down** from multi-user to single-user. If you wish, you can mount any combination of file systems in single-user mode. Ensure that everything is unmounted before going multi-user.
3. You should always unmount the user file system when backing it up to tape to prevent access/update to the file system.
4. If your user directories are located in a mountable file system, you can deny login access to those directories by unmounting it. Some administrators group users into different file systems for access/denial control. Since the system cannot find their directory, they will get a No directory message if they try to log in.

---

## File System Status

The **df** command tells you how much data space and free i-nodes are available on those mounted file systems registered in the mount table **/etc/mnttab**. Executing **df** with the **-t** flag (**df -t**) gives you a total listing, which includes available free space and total block space per file system. This command can be run by any user on the system and should be executed often to ensure that you have enough available space to accommodate file growth. A sample **df** listing is shown below on a system with two additional file systems mounted:

```
/apl (/dev/dsk/0s6 ): 44780 blocks 13778 i-nodes
/u (/dev/dsk/0s3 ): 44165 blocks 11556 i-nodes
/ (/dev/dsk/0s1 ): 8474 blocks 3422 i-nodes
```

The **mount** command, with no arguments, will also tell you what is mounted according to the **/etc/mnttab** file in the format below:

```
/ on /dev/dsk/0s1 @ [local] read/write on Tue May 30 08:11:56 1986
/u on /dev/dsk/0s3 @ [local] read/write on Tue May 30 08:11:57 1986
/apl on /dev/dsk/0s6 @ [local] read/write on Tue May 30 08:11:58 1986
```

The information displayed by **df** and **mount** is only as valid as the data stored in the **/etc/mnttab** file. Inaccurate mount information can be displayed by these commands. For example, **df** can show two entries for a single file system indicating it is mounted more than once. This is physically impossible. Or a file system might not show up at all, reflecting that it is unmounted, but **ls** on the mountstub directory reveals file and directory names, which indicate that it truly is mounted. These conditions are an indication that the **mnttab** file is in error. Reasons for the **mnttab** file to be in error include:

- the **mnttab** file was not reset on reboot.
- the system is still in single-user mode after a crash and **/etc/brc** has not had a chance to clear and redefine the table on the way to multi-user mode.

The **/etc/mnttab** file should be corrected by changing system states (i.e. single-user to multi-user) or by using the **setmnt** command directly.

---

## Using the Ramdisk Driver

The Plexus driver `/usr/plx/ramdisk` uses system memory as a disk. This utility allocates memory to one of up to eight devices. The directory `/dev/rram` contains raw devices, which are mostly used as parameters to the utility. The directory `/dev/ram` contains block devices which can be configured to be mounted file systems.

In order to use the `ramdisk` driver effectively, you should be an experienced system administrator and understand how memory is allocated for your Plexus system. Allocating too much memory as a disk can cause your system's resources to be taxed which might slow down response time. If the demands on the system become extensive, the system might be halted. Reaching the limits of your system's memory capacity manifests itself as thrashing. See the section "Thrashing" in the chapter *Managing System Processes* for additional information.

The simplest way to allocate system memory as a disk is to uncomment (remove the # from) the appropriate portion of the `/etc/rc` file. All of the necessary commands to use system memory as a disk are in `/etc/rc`. See the section "Entering Multi-User Mode" in the chapter *Startup and Shutdown* for a listing of the `/etc/rc` file.

Note that using `ramdisk` on certain file systems, such as `/lib`, creates a read-only ram disk. Thus, you cannot install software products in `/lib` when `/lib` is mounted as a read-only ram disk. To install software in `/lib`, you must bring the system down to single-user mode, install the software products in single-user mode, then bring the system back up to multi-user mode. See the chapter *Startup and Shutdown* for more information.

For more information on the `ramdisk` driver see: `ramdisk(1M)`, `rram(7)`, and `ram(7)` in the *Sys5 UNIX Administrator's Reference Manual*.

## CONFIGURING DISKS

---

This chapter explains how to configure the remainder of your disk (after `/dev/dsk/0s2` - the swap area). It also explains how to configure a second drive. This chapter assumes you will be using the Plexus default root file system (`/dev/dsk/0s1`) and swap area (`/dev/dsk/0s2`).

If you need to increase the root file system size or increase or decrease the swap area size, see the chapter *Changing Root and Swap*. Perform the procedures in the chapter *Changing Root and Swap* BEFORE setting up additional file systems described in this chapter.

The 1Gb 12-inch optical disk is also available from Plexus. This disk is configured at the factory. For additional information see the *Plexus Optical Disk User's Manual*.



---

## Introducing **dconfig**, **mkfs**, and **mknod**

Configuring disks for a Plexus system involves various Sys5 utilities. Some of these utilities include: **dconfig**, **mkfs**, and **mknod**. This section introduces you to these utilities.

### Using **dconfig**

**dconfig** is a Plexus stand-alone utility. It defines the disk space by logical disks allocated in 512-byte sectors. You must run **dconfig** on the file system containing your system files. By default, this is **/dev/dsk/0s1**.

The procedures in this chapter are based on Plexus systems with one type of disk controller. Some Plexus systems, such as P/60s and P/75s, have IMSP and EMSP controlled disks in the same system. In systems with IMSP and EMSP controllers, **/dev/dsk/Xsxx** defaults to the EMSP controller. To access disks controlled by the IMSP, use **/dev/pd/Xsxx**. Read the section "Systems With IMSP and EMSP Controllers" in the chapter *Overview of Sys5 File Systems* before configuring your disk.

You can run **dconfig** to obtain or change any of the following parameters:

- Disk identification (id)
- Default kernel name (primary bootname)
- Alternate kernel name (secondary bootname)
- Disk information specific to the actual disk unit installed, including number of cylinders, sector size, sectors per track, etc.
- System's node name (if you want to give your system a name to put it on a **uucp** network)
- Which logical disk will contain the root file system and perform pipes and dumps
- Which logical disk will provide the swap area

- 
- Address and size of the swap area
  - Boundaries and start addresses of each potential logical disk

The **dconfig** parameters are vital to the system. Some parameters (such as disk cylinders and sector size) have no additional values for your system. **CHANGING THEM SIMPLY DISABLES YOUR SYSTEM.** Other parameters can be changed if done carefully and consistently, such as: device numbers for root and swap, swap size and location, starting addresses of the logical disks.

**[ NOTE ]** Use **dconfig** to make changes when in stand-alone mode at the boot ( : ) prompt (i.e. Sys5 is not booted). This ensures that the disk is properly set up for your needs before you configure any root files.

Thus, the Plexus utility, **dconfig**, can alter the fundamental organization of your disk(s). You should run stand-alone **dconfig** before booting Sys5 under any of the following circumstances:

- You want your root file system larger or smaller than the capacity as shipped from the factory.
- Your system needs a swap area larger or smaller than the swap area set up by Plexus at the factory.
- You want your swap area moved to a different logical disk. This is especially applicable for older systems shipped with the swap space on `/dev/dsk/0s1`.
- You want to establish or change the definition of any or all logical partitions on the disk.

---

YOU CANNOT incorporate the first two sectors on a disk (i.e., sector 0 and sector 1) in a logical file system other than the first logical file system. The descriptor which refers to the physical disk (0s0, 1s0, 2s0, ...15s0) MUST incorporate the first two sectors (sector 0 and sector 1) of the disk. Thus, **/dev/dsk/0s0** MUST be configured to begin at sector 0.

The descriptor which refers to the first logical file system (i.e., 0s1, 1s1, 2s1, ... 15s1) can (but does not have to) incorporate the first two sectors (sector 0 and sector 1) of the disk. Thus, **/dev/dsk/0s1** can be configured to begin at sector 0.

ALL OTHER DESCRIPTORS which refer to the other logical file systems on a disk (0s2 ... 0s15, 1s2 ... 1s15, and so on) MUST NOT begin at sector 0 or 1. Also, a logical file system MUST begin on an even sector.

If a logical disk is used as a raw disk for a data base (i.e. it is not used as a file system), it must start after block 0 (sector 0 and sector 1 comprise the first block).

## Using mkfs

**mkfs** is a Sys5 utility which sets up file systems in 1K (actually, 1024 bytes) blocks which are integral to the operation of Sys5.

**[ NOTE ]** When configuring disks, remember that **mkfs** uses 1024-byte blocks while **dconfig** uses 512-byte sectors.

When creating additional file systems you can run either **/etc/mkfs** or stand-alone **mkfs**. **/etc/mkfs** is run when Sys5 is booted. However, stand-alone **mkfs** must be run from the boot (:) prompt (before Sys5 is booted). When creating the root file system, stand-alone **mkfs** MUST BE used because it prepares a file system so Sys5 and its utilities (the root file system) can be restored from tape onto the disk.

---

The stand-alone program, **mkfs**, and its **/etc/mkfs** counterpart, establishes file system structure on a predefined logical disk. It declares that a certain range of disk addresses is to be considered a file system, and sets up disk blocks (free list, superblock, the i-list) within that file system for Sys5 to maintain housekeeping information about the files within the file system. The superblock, which contains information such as how many files are in the file system, how many blocks are free, where files live on the disk, and how big they are, is also defined.

### Using mknod

When a new device is added to your Plexus system, it is sometimes necessary to add a new device name in the **/dev** directory. This is called making a node and is done with the **mknod** command. **mknod** specifies the logical disks, with their start addresses listed in **dconfig**, to be created and utilized in Sys5's device directory, **/dev**.

Plexus systems are shipped with Sys5 loaded and operable on **/dev/dsk/0s1**. Swap space is set up by Plexus on **/dev/dsk/0s2**.

**[ NOTE ]** DO NOT invoke **mknod** to create nodes for the first logical disk (**0s0 - 0s15**) and **1s0** and **2s0**. Plexus has created these nodes in **/dev** for you.

---

## Creating Additional File Systems

In order to set up additional file systems, you must further delineate the disk space between the end of `/dev/dsk/0s2` (swap) and the end of the disk. A few examples of additional file systems you may want to include on your first physical disk might be a file system for user files (`/user`) and a file system for programming languages (`/lang`).

It is advisable, but not necessary, to keep each additional file system at least slightly smaller than your tape backup capacity. For reel-to-reel tape, you can archive approximately 1 Mb for every 60 feet of tape if you are using a low-overhead utility such as **dump** or **fbackup**. At this rate, you could get up to 40 Mb on a 2400-foot reel of tape (if your tape is in good condition). Cartridge tape drives have a 45Mb (P/35) or 60Mb (P/55, P/75, P/90, and P/95) capacity in streaming mode (not in **dump** or **cpio** mode). Older P/35 cartridge tape drives have a 20Mb capacity.

An advantage of smaller file systems is faster disk I/O because the operating system does not have to search through an extremely large disk area. The disadvantage is that it requires more **dump** or **fbackup** operations to backup the entire disk.

---

The following table lists Plexus' current **dconfig** default for all disk drives except the 22Mb and 36Mb 8" IMSP drive. Note that the information is in 512-byte sectors. The ~ means end of disk.

**Table 8-1.** Sample **dconfig** Display in 512 Byte Sectors

dsk/0s0	[0,0]:	0,~
dsk/0s1	[0,0]:	0,68000
dsk/0s2	[0,0]:	68000,32000
dsk/0s3	[0,0]:	100000,~
dsk/0s4	[0,0]:	140000,~
dsk/0s5	[0,0]:	180000,~
dsk/0s6	[0,0]:	220000,~
dsk/0s7	[0,0]:	260000,~
dsk/0s8	[0,0]:	300000,~
dsk/0s9	[0,0]:	400000,~
dsk/0s10	[0,0]:	500000,~
dsk/0s11	[0,0]:	600000,~
dsk/0s12	[0,0]:	700000,~
dsk/0s13	[0,0]:	800000,~
dsk/0s14	[0,0]:	900000,~
dsk/0s15	[0,0]:	1000000,~

The **dconfig** values in the previous table are sample values only. Consult your latest release notice for current **dconfig** values. The symbol ~ means to the end of the disk. Knowing what the end is (the sector count for your disk minus the starting sector address for that logical disk) enables you to make full use of your disk when creating file systems with **mkfs**. You will need to run stand-alone **dconfig** to verify the Plexus default when told to do so.

The root file system and swap space have been set up at the factory using Plexus' default **dconfig** standards. It is up to you to configure the remainder of your disk (0s3 - 0s15). It is suggested to use the default values for root (0s1) and swap (0s2) set up by Plexus. You can look at these values by running stand-alone **dconfig**.

---

## Calculating Total Sectors Available

To configure the remainder of your disk (after the swap space, /dev/dsk/0s2), you must know the total sectors available (TSA). Table 8-2 lists the total sectors available in 512-byte sectors and 1024-byte blocks for Plexus IMSP, EMSP, AMSP (P/95), and MMSP (P/90) disks.

**Table 8-2.** Total Sectors and Blocks Available

Disk Size	Controller Name	Total 512-byte Sectors	Total 1024-byte Blocks
NEC 22Mb 8"	IMSP	40046	20043
NEC 36Mb 8"	IMSP	66810	33405
Fuj. 71Mb 8"	IMSP	134946	67473
Fuj. 72Mb 14"	IMSP	136170	68085
NEC 142Mb 8"	IMSP	272544	136272
Fuj. 145Mb 8"	IMSP	273020	136510
Fuj. 145Mb 14"	IMSP	272340	136170
NT 270Mb 8"	IMSP	518760	259380
Fuj. 285Mb 14"	IMSP	545088	272544
Fuj. 71Mb 8"	EMSP	131488	65744
Fuj. 72Mb 14"	EMSP	135465	67732
NEC 142Mb 8"	EMSP	261632	130816
Fuj. 145Mb 8"	EMSP	263360	131680
Fuj. 145Mb 14"	EMSP	270930	135465
Fuj. 285Mb 14"	EMSP	539616	269808
NEC 286Mb 8"	EMSP	541860	270930
NEC 443Mb 9"	EMSP	835316	417658
Fuj. 570Mb 14"	EMSP	1079232	539616
Fuj. 586Mb 10.5"	EMSP	1108800	554400
Fuj. 145Mb 8"	AMSP	269940	134970
Fuj. 286Mb 8"	AMSP	548060	274030
NEC 443Mb 9"	AMSP	849010	424505
NEC 142Mb 8"	AMSP	252402	126201
Fuj. 67Mb 5 ¼"	MMSP	141225	70612
Fuj. 135Mb 5 ¼"	MMSP	282450	141225
Max. 300Mb 5 ¼"	MMSP	507360	253680
Fuj. 310Mb 5 ¼"	MMSP	637875	318937

---

The total number of 512-byte sectors available on the P/35, P/55, P/60, P/75, P/90, and P/95 systems is calculated using the following equation:

$$\text{TSA} = ((\text{NCYL} - \text{ALTCYL} - \text{TSTCYL}) * \text{NHEADS}) - \text{ALTTRKS}) * (\text{NSECTPT} - \text{ALTSECT})$$

Where:

TSA	total number of sectors available
NCYL	number of cylinders on the drive (see Tables 8-3 through 8-6)
ALTCYL	number of alternate cylinders allocated for the remapping of bad tracks (IMSP=20, EMSP and AMSP=0, MMSP=13)
TSTCYL	number of test cylinders reserved (IMSP=2, EMSP and AMSP=0, MMSP=3)
NHEADS	number of heads on the drive (see Tables 8-3 through 8-6)
ALTTRKS	Number of alternate tracks allocated for the remapping of bad tracks (AMSP=50, all others=0)
NSECTPT	total number of sectors per track (see Tables 8-3 through 8-6)
ALTSECT	number of alternate sectors per track allocated for the slipping of bad sectors (IMSP and MMSP=0, AMSP=1, EMSP=2)



Tables 8-3 through 8-6 list disk configuration values.

**Table 8-3. AMSP Disk Configuration Values**

<b>Disk Size</b>	<b># of Cyl.</b>	<b># Fixed Heads</b>	<b>Alt. Tracks</b>	<b>Sect./ Track</b>	<b>Sect./ Track (not incl. spares)</b>
145Mb 8"	823	10	50	34	33
142Mb 8"	1024	8	50	32	31
286Mb 8"	823	10	50	68	67
443Mb 9"	760	19	50	60	59

**Table 8-4. MMSP Disk Configuration Values**

<b>Disk Size</b>	<b># of Cyl.</b>	<b># Fixed Heads</b>	<b>Sectors/ Track</b>
67Mb 5 ¼"	823	5	35
135Mb 5 ¼"	823	10	35
300Mb 5 ¼"	1224	15	28
310Mb 5 ¼"	1231	15	35

**Table 8-5. IMSP Disk Configuration Values**

<b>Disk Size</b>	<b># of Cyl.</b>	<b># Fixed Heads</b>	<b>Sectors per track</b>
22Mb 8"	415	3	34
36Mb 8"	415	5	34
71Mb 8"	589	7	34
72Mb 14"	823	5	34
142Mb 8"	1024	8	34
145Mb 8"	823	10	34
145Mb 14"	823	10	34
270Mb 8"	1332	12	33
285Mb 14"	1024	16	34

---

**Table 8-6. EMSP Disk Configuration Values**

<b>Disk Size</b>	<b># of Cyl.</b>	<b># Fixed Heads</b>	<b>Bytes/ Sect</b>	<b>Sectors/ Track</b>	<b>Sect/track not incl. spares</b>
71Mb 8"	589	7	590	34	32
72Mb 14"	823	5	582	35	33
142Mb 8"	1024	8	586	34	32
145Mb 8"	823	10	n/a	34	32
145Mb 14"	823	10	582	35	33
285Mb 14"	1024	16	582	35	33
286Mb 8"	823	10	592	68	66
443Mb 9"	760	19	604	60	58
570Mb 14"	1024	16	598	68	66
586Mb 10.5"	842	20	600	68	66

TSA is used to calculate the end of the each logical disk. For example, assume that you have a 142Mb 8" IMSP disk. Your root file system (0s1) is 34Mb (68000 512-byte sectors); your swap space (0s2) is 16Mb (32000 512-byte sectors). Referring to Table 8-2, the total number of 512-byte sectors available on your disk is 272544. The root file system takes up 68000 512-byte sectors (34000 X 2) and the swap space takes up 32000 512-byte sectors (16000 X 2) totaling 100000 512-byte sectors. The total sectors available after `/dev/dsk/0s2` is calculated by subtracting the number of 512-byte sectors for root and swap from the TSA:

TSA - (sectors used for 0s1 and 0s2) = total 512-byte sectors remaining

272544 - 100000 = 172544 (512-byte sectors remaining)

This means you have 172544 available sectors for creating one or more additional disk file systems.

---

## Cylinder Size

**dconfig** cylinder size (cylsize) is for calculating starting addresses of logical disks. System performance is improved if logical disks start on a cylinder boundary. Starting sectors of cylinders are multiples of the **dconfig** cylsize (sectors per cylinder).

Following is the formula for calculating the cylinder size for **dconfig**:

$$\text{DCYLSIZE} = \text{NHEADS} * (\text{NSECTPT} - \text{ALTSECT})$$

DCYLSIZE = number of sectors per cylinder

The cylinder size to use for **mkfs** can be calculated with the following formula:

$$\text{MCYLSIZE} = (\text{NHEADS} * (\text{NSECTPT} - \text{ALTSECT})) / 2$$

Where:

**NHEADS** the number of heads on the disk drive (see Tables 8-3 through 8-6).

**NSECTPT** the total number of sectors per track (see Tables 8-3 through 8-6).

**ALTSECT** the number of alternate sectors per track allocated for the slipping of bad sectors (IMSP and MMSP = 0, EMSP = 2).

## Step-by-Step Disk Configuration Overview

The procedure described in this subsection provides an overview of the steps needed to configure the remainder of your disk. A step-by-step example follows the overview.

Thus, the basic sequence for configuring a disk requires you to:

1. Determine how you want to divide your disk(s). Each physical disk can contain up to 16 logical disk partitions.
2. Run **fsck** on ALL file systems.

- 
3. Back up the entire disk before executing **dconfig** to avoid losing your data.
  4. At the boot prompt (:) run stand-alone **dconfig** to ensure that **dconfig**'s default values are sufficient for your goals. If the default values do not meet your needs, change them, specifying the start addresses and sizes in 512-byte sectors.
  5. After exiting stand-alone **dconfig**, boot Sys5 to single-user mode.
  6. Check the **/dev** directory for entries for each logical disk. If any are missing, create them with the **mknod** command. These logical devices must correspond to the start addresses you picked or established with **dconfig**. Set the read/write/execute permissions (only root can read) and group ownership of the newly created devices.
  7. Make the file systems with the **mkfs** command. This assigns the file systems to the logical disks you created, and determines the file system size in 1K blocks.
  8. Edit **/etc/checklist** to add to the list all file systems to be checked.
  9. Run **fsck** to check the file systems. Running **fsck** at this point will catch errors that might otherwise be missed until the data is loading and/or the procedure is completed.
  10. Create or assign Sys5 directories to the logical disks/file systems you just created with **mkdir**.
  11. Name the file systems with **/etc/labelit** (this step is optional). The file systems should be unmounted to use this command, which eliminates the warning messages generated by the **mount** command. **/etc/labelit** is documented in the *UNIX Administrator Reference Manual* under **volcopy(1M)**. The **labelit** command will be briefly explained in the subsection *Creating Mounted File Systems*, later in this chapter.
  12. Mount the file systems to the directories with **mount**.

- 
13. To verify that the file systems are set up correctly, unmount the file systems then mount them again. If you receive an error message after unmounting the file systems, the file systems are not set up correctly.
  14. Create a **lost+found** directory in the mountstub directory root of each file system. Note that the file system must be mounted for this step.
  15. Edit the `/etc/rc` file and add your `/etc/mount` and `/etc/umount` commands.
  16. Bring the system up to multi-user mode then verify that the file systems are mounted.

### Step-by-Step Disk Configuration Example

In the following example, assume you have a Plexus system with an 8-inch 142Mb IMSP disk drive. File systems for `/dev/dsk/0s1` (root) and `/dev/dsk/0s2` (swap) are already set up by Plexus. This example assumes you want to create two additional file systems beginning with `/dev/dsk/0s3`.

**[ NOTE ]** DO NOT follow this procedure verbatim. It is provided to illustrate the facets of setting up your own file systems, especially for calculating the sizes of mounted file systems.

1. To begin, determine how you want to split up your disk. To do this, you need the total sectors available on your drive. In this example we will configure the 142Mb 8" IMSP drive. Referring to Table 8-2, note that the total 512-byte sectors available is 272544. This is the starting point for your calculations.

2. Draw a sketch illustrating how you want your physical disk subdivided into logical disks, similar to Figure 8-1, below.

0s1	sector 0 34Mb root file system (68000 sectors)
0s2	sector 67999 16Mb swap space (32000 sectors)
0s3	sector 99999 20Mb file system (40000 sectors) /user
0s4	sector 139999 30Mb file system (60000 sectors) /lang
0s5	sector 199999 ??Mb file system
.	.
0s15	272544 total sectors available

**Figure 8-1.** Sample Disk Organization (512-byte sectors)

3. By looking at Table 8-2 and Figure 8-1, note that `/dev/dsk/0s1` is 34Mb (68000 512-byte sectors). The starting address is 0 and the number of sectors is 68000. `/dev/dsk/0s2` is 16Mb (32000 sectors), extends to 100000 512-byte sectors (68000 + 32000).

- 
4. Assume that you want the Plexus default size (20Mb) file system for `/dev/dsk/0s3`. Table 8-1 shows that the starting address of `/dev/dsk/0s3` is 100000 512-byte sectors. `/dev/dsk/0s3` will extend the number of sectors used to an additional 40,000 sectors.
  5. Assume that you want a 30Mb file system on `/dev/dsk/0s4`. The beginning address is 140000 and the number of sectors is 60000. Through the end of `/dev/dsk/0s4`, we have now used 200000 sectors of the disk (sector 0 through sector 199999). Remembering that the TSA (in 512-byte sectors) is 272544, subtract 199999 from TSA:

$$272544 - 199999 = 72545$$

The resulting value, 72545, is the total number of 512-byte sectors the remainder of the disk can occupy (0s5 - 0s15).

**mkfs** uses 1K byte blocks. To obtain the number of 1K byte blocks, divide 72545 by 2.

$$72544 / 2 = 36272$$

This is approximately 36.2Mb of available disk after considering 0s1, 0s2, 0s3, and 0s4.

6. After determining how you want your disk divided, shut the system down to single-user mode. See the chapter *Startup and Shutdown*.
7. Run a file system check on ALL file systems. The **fsck** command performs a file system check on all file systems that are listed in the file `/etc/checklist`, so you must ensure that all file systems that you want checked are listed in this file. Use a text editor to add any missing file systems. To run **fsck**, enter:

**fsck**

- 
8. Back up the root file system using **/etc/dump**. Enter:

**/etc/dump 0 /dev/xxx/0m /dev/dsk/0s1**

Where *xxx* is:   **rrm** for reel-to-reel tape [raw device]  
                  **rpt** for cartridge tape [raw device]  
                  **crm** for caching reel-to-reel tape  
                  (Multibus systems only)  
                  **rft** for caching cartridge tape (Mul-  
                  tibus systems only)

9. Back up all remaining file systems using **cpio**.

DO NOT CONTINUE UNTIL YOU HAVE BACKED UP ALL FILE SYSTEMS.

The **cpio** process involves mounting the file system, changing to the mountstub directory, invoking **cpio** to back up the file system, checking your tape with the **-t** option, and unmounting the file system. Enter each of the following, in turn:

- a. **mount /dev/dsk/0s3 /mountstub3** (i.e. **/user**)
- b. **cd /mntstub3** (i.e. **/user**)
- c. **find . -print -cpio /dev/xxx/0m**

Where *xxx* is:   **rpt** for cartridge tape [raw device]  
                  **rrm** for reel-to-reel tape [raw device]  
                  **rft** for caching cartridge tape (Mul-  
                  tibus systems only)  
                  **crm** for caching reel-to-reel tape (Mul-  
                  tibus systems only)

- d. **cpio -itB </dev/dsk/0s3**
- e. **umount /dev/dsk/0s3**
- f. **mount /dev/dsk/0s4 /mntstub4** (i.e. **/lang**)
- g. **cd /mntstub4** (i.e. **/lang**)



---

h. **find . -print -cpio /dev/xxxx/0m**

Where *xxx* is     **rpt** for cartridge tape [raw device]  
                  **rrm** for reel-to-reel tape [raw device]  
                  **rft** for caching cartridge tape (Multi-  
                  bus systems only)  
                  **crm** for caching reel-to-reel tape (Multi-  
                  bus systems only)

i. **cpio -itB </dev/dsk/0s4**

j. **umount /dev/dsk/0s4**

Continue the **mount**, **cd**, check **cpio** tape, and **umount** procedure for all remaining file systems.

10. Reset the system. You can press RESET, enter **sys reset** (68020 systems only), or turn the power on key off and back on. The system goes through a self-test. At the boot (:) prompt, run stand-alone **dconfig**. Enter:

**dconfig**

11. At the \$\$ **dconfig** prompt enter:

**/dev/dsk/0s0**

Sample **dconfig** displays are presented on the following pages (see Figures 8-2 through 8-5).

In the **dconfig** displays, press **<cr>** after each response.

**[ NOTE ]** If you have a system with both IMSP and EMSP disk controllers, you must enter the nodename (i.e. the system name) of your system at the **sys5 nodename** prompt before pressing **<cr>**.

---

```
:dconfig
$$ dconfig /dev/dsk/0s0 <cr>
Disk id? [pd]: <cr>
Primary bootname? [/unix]: <cr>
Secondary bootname? []: <cr>
Number of cylinders? [*]: <cr>
Number of heads on removable? [0]: <cr>
Number of heads on fixed? [*]: <cr>
Data bytes per sector? [*]: <cr>
Sectors per track? [*]: <cr>
Number of alternate cylinders? [*]: <cr>
File system blocksize? [*]: <cr>
Sys5 nodename? []: <6 character system node name> <cr>
Change the default unix device mapping? [y/n]: y <cr>
```

**Figure 8-2. dconfig Display, IMSP Drives**

```
:dconfig
$$ dconfig /dev/dsk/0s0 <cr>
Disk id? [xy]: <cr>
Primary Bootname? [/unix]: <cr>
Secondary Bootname? []: <cr>
Disk Type: 0
Number of cylinders? [*]: <cr>
Number of heads on fixed? [*]: <cr>
Sectors per track not including spares? [*]: <cr>
Number of spare sectors? [*]: <cr>
Nodename? []: <6 character system node name> <cr>
Change the default unix device mapping? [y/n]: y <cr>
```

**Figure 8-3. dconfig Display, EMSP Drive**

- \* These numbers depend on the drive type. They are selected during **dformat**.

---

```

:dconfig
$$ dconfig /dev/dsk/0s0 <cr>
Disk id? [sm]: <cr>
Primary bootname? [/unix]: <cr>
Secondary bootname? []: <cr>
Disk Type: 0
Number of cylinders? [*]: <cr>
Number of heads on fixed? [*]: <cr>
Sectors per track not including spares? [*]: <cr>
Number of spare sectors 1
Number of alternate tracks? [*]: <cr>
Nodename? []: <6 character node name> <cr>
Runt sector enable? [*]: <cr>
Gap 2? [*]: <cr>
Change the default unix device mapping? [y/n]: y <cr>

```

**Figure 8-4. dconfig Display, AMSP Drive**

```

:dconfig <cr>
: sd/0s0 137 MB host=0 sid=6 lun=0 size=282450
$$ dconfig /dev/dsk/0s0 <cr>
Disk id? [sd]: <cr>
Primary bootname? [/unix]: <cr>
Secondary bootname? []: <cr>
Number of cylinders? [*]: <cr>
Number of heads on fixed? [*]: <cr>
Data bytes per sector? [*]: <cr>
Sectors per track? [*]: <cr>
Number of controller reserved cylinders 3
Number of alternate cylinders? [*]: <cr>
Nodename? []: 6 character node name> <cr>
Change the default unix device mapping? [y/n]: y <cr>

```

**Figure 8-5. dconfig Display, MMSP Drive**

- \* These numbers depend on the drive type. They are selected during **dformat**.

- 
12. After entering **y** at the Change the default unix device mapping? [y/n]: prompt, the display continues. For this example, the root file system is on **/dev/dsk/0s1** and the 16Mb swap area is on **/dev/dsk/0s2**.

Rootdev	1
Pipedev	1
Dumpdev	2
Swapdev	2
Swaplo	0
Nswap	32000

Following is an explanation of the display.

Rootdev      If **1** displays, press **<cr>**. If **1** does not display, enter **1**. Rootdev specifies the device where the root file system is located (**/dev/dsk/0s1**).

Pipedev      If **1** displays, press **<cr>**. If **1** does not display, enter **1**. Pipedev specifies where pipes are to be allocated. This must be a mounted file system and must be the same as rootdev.

Dumpdev      If **2** displays, press **<cr>**. If **2** does not display, enter **2**. Dumpdev specifies the device used to dump memory after a system crash. Default is **/dev/dsk/0s2**.

Swapdev      If **2** displays, press **<cr>**. If **2** does not display, enter **2**. Swapdev specifies the file system containing the swap area. It must be the same as dumpdev.

Swplo        If **0** displays, press **<cr>**. If **0** does not display, enter **0**. Swaplo specifies the starting address (in 512-byte sectors) of the swap area within a logical partition. Since the swap area comprises the entire **0s2** partition, a value of **0** means to start at the beginning of logical disk **/dev/dsk/0s2**.

---

Nswap            If **32000** displays, press **<cr>**. If **32000** does not display, enter **32000**. Nswap is the number of 512-byte sectors (i.e. 16000 X 2) to be used for the swap area starting at swaplo.

13. Next the system displays:

Change the file system disk configuration? [y/n]:

Enter **y** and the first 16 lines in Figure 8-6 display. If the figures shown in **bold** (as user input) appear in the column that now contains *[0,0]*, and are correct for your desired configuration, press **<cr>**. If the column contains an incorrect figure or zeroes, enter the desired number and press **<cr>**.

The first number in the right-hand column specifies the beginning address of the file system (in 512-byte sectors). The number after the comma specifies the length of the file system in 512-byte sectors. For example, the root file system (0s1) consists of 68000 sectors; it starts at sector 0 and extends through sector 67999 (see Figure 8-1).

```

dsk?s0 [0,0]: 0,~
dsk?s1 [0,0]: 0,68000
dsk?s2 [0,0]: 68000,32000
dsk?s3 [0,0]: 100000,40000
dsk?s4 [0,0]: 140000,60000
dsk?s5 [0,0]: 200000,~
dsk?s6 [0,0]: <cr> †
dsk?s7 [0,0]: <cr>
dsk?s8 [0,0]: <cr>
dsk?s9 [0,0]: <cr>
dsk?s10 [0,0]: <cr>
dsk?s11 [0,0]: <cr>
dsk?s12 [0,0]: <cr>
dsk?s13 [0,0]: <cr>
dsk?s14 [0,0]: <cr>
dsk?s15 [0,0]: <cr>

Is the above information correct? [y/n]: y
Are you sure you want to rewrite block 0?: y‡
Block 0 of [pd](0,0) initialized successfully!
Exit n (n is 0 if yes, 1 if no)
PLEXUS PRIMARY BOOT REV X.X
:

```

**Figure 8-6.** Sample `dconfig` Display

**[ NOTE ]** The number after the comma designates the size of the partition. It DOES NOT mean the ending sector of the partition.

† When you have configured as many drives as you want, you can respond with a `q <cr>` to the next question and quit from this program, rather than press a `<cr>` through each of the 16 inquiries.

‡ If you are verifying data only, answer `n` to this prompt.

- 
14. After exiting **dconfig**, boot Sys5 by pressing **<cr>** at the boot prompt.
  15. Check the device directory, **/dev**, for entries for each logical disk. Enter:

```
ls -l /dev
```

As shipped from the factory, your device directory already contains logical disks **0s0** through **15s0**. You must set up **Xs1 - Xs15** for any additional drives in **/dev/dsk** and **/dev/rdisk**. If the appropriate logical disk is not listed in **/dev**, execute **mknod**. Because **0s3** and **0s4** were listed in **/dev**, you DO NOT have to create these devices in **/dev**. However, the **mknod** syntax follows for your information.

```
/etc/mknod <devname> <char or block> <maj dev#> <minor dev#>
```

[ **NOTE** ] The following is a sample of using **mknod** with specific device names. DO NOT enter the commands listed below if **0s3** and **0s4** are listed in **/dev/dsk**.

```
/etc/mknod /dev/dsk/0s3 b 0 3  
/etc/mknod /dev/dsk/0s4 b 0 4  
/etc/mknod /dev/rdisk/0s3 c 3 3  
/etc/mknod /dev/rdisk/0s4 c 3 4
```

Where: **b** stands for block device.  
**c** stands for character device.  
**0** is the block major device number for **dsk**.  
**3** is the character major device number for **rdsk**.  
**3** and **4** are the minor device numbers.

You will use the same syntax to set up each logical disk (only the minor device number will change). Table 8-7 lists the minor device numbers used in **mknod** for the first three physical drives.

---

**Table 8-7. Minor Device Numbers Used by `mknod`**

<b>Logical Disk</b>	<b>Physical Disk 0 0s</b>	<b>Physical Disk 1 1s</b>	<b>Physical Disk 2 2s</b>
0	0	16	32
1	1	17	33
2	2	18	34
3	3	19	35
4	4	20	36
5	5	21	37
6	6	22	38
7	7	23	39
8	8	24	40
9	9	25	41
10	10	26	42
11	11	27	43
12	12	28	44
13	13	29	45
14	14	30	46
15	15	31	47

16. Set the read/write/execute permissions (only root can read) and group ownership of the new devices. Enter:

```
cd /dev/dsk
chmod 400 0s3 0s4
chgrp sys 0s3 0s4
cd /dev/rdisk
chmod 400 0s3 0s4
chgrp sys 0s3 0s4
```



- 
17. To create a file system, execute the **mkfs** command. **mkfs** assigns file systems (in 1K blocks) to the logical disks created with **mknod**.

[ **NOTE** ] **mkfs** uses 1K blocks, not 512-byte sectors, for establishing the size of a file system partition. Divide the sector count of the partition by two to get the size in 1K blocks.

Enter:

```
/etc/mkfs /dev/dsk/0s3 20000 softint cylsize  
/etc/mkfs /dev/dsk/0s4 30000 softint cylsize
```

Where: **/dev/dsk/0s3** is the special file name  
**20000** is the number of 1K blocks  
**softint** is the software interleave  
**cylsize** is blocks/cylinder

18. Edit the file **/etc/checklist** using a text editor. Ensure that all file systems are listed in this file.
19. Run **fsck** on each new file system to insure each file system's integrity. Enter:

```
fsck
```

See **fsck(1M)** in the *Sys5 UNIX Administrator Reference Manual* and the chapter *Stand-alone Programs* for additional information on **fsck**.

20. After running **fsck**, sync the system. Enter:

```
sync
```

21. Create or assign Sys5 directories to the logical disks/file systems you just created via **mkdir**. In the example below, the user and language directories are created. Enter:

```
/etc/mkdir /user  
/etc/mkdir /lang
```

- 
22. Name the file systems with **labelit**. This step is optional but will prevent warning messages from occurring when the file system is mounted without a label. See the *UNIX Administrator Reference Manual* under **volcopy(1M)** for a complete description of this command. Note that the file systems should be unmounted to use this command. The assigned name (label) will be overwritten when the file system is unmounted. Enter:

```
/etc/labelit /dev/dsk/0s3 user 0s3
/etc/labelit /dev/dsk/0s4 lang 0s4
```

Where: **/dev/dsk/0s3** is the special file name (special files are devices)  
**user** is the directory where the file system is mounted — no leading /  
**0s3** is the volume name

23. Mount the logical block devices to the directories you created or chose in the last step:

```
/etc/mount /dev/dsk/0s3 /user
/etc/mount /dev/dsk/0s4 /lang
```

24. After mounting the file systems, ensure that the file systems are mounted and set up correctly. Unmount the file systems, then mount them again. If you cannot unmount the file systems without receiving an error message, the file systems are not set up correctly.

```
/etc/umount /dev/dsk/0s3
/etc/umount /dev/dsk/0s4
/etc/mount /dev/dsk/0s3 /user
/etc/mount /dev/dsk/0s4 /lang
```

- 
25. Set up a **lost+found** directory in each file system you created. To set up **lost+found** you will create a **lost+found** directory, copy files from **/bin** to your newly created **lost+found** directory in the root of the file system, list the files to make sure they were copied correctly, and remove the files, thus creating a large, empty space in **lost+found**. To set up **lost+found** in the **user** directory enter:

```
mkdir /user/lost+found
cp /bin/* /user/lost+found
ls -C /user/lost+found
rm /user/lost+found/*
```

Your **lost+found** directory in **user** is now set up. To set up a **lost+found** directory in the **lang** directory, repeat the above steps replacing **user** with **lang**.

26. Edit the **/etc/rc** file and add the **/etc/umount** and **/etc/mount** commands for the file systems (**/dev/dsk/0s3** and **/dev/dsk/0s4**) that you established in the previous steps. Figure 8-7 shows the portion of the **/etc/rc** you will alter. The lines to be added are in bold.
27. Bring the system up into multi-user mode. Enter:
- ```
init 2
```
28. **/etc/rc** should mount your file systems. Verify this with the **mount** command.

---

```

#    @(#)Mrc.sh 1.6

TZ=PST8PDT
export TZ

if [ ! -f /etc/mnttab ] ; then
    /etc/devnm / | grep -v swap | grep -v root | /etc/setmnt
fi
set who -r
CurrentState=$7
TimesInCurrentState=$8
PreviousState=$9

case $CurrentState in
S|1)
    if [ $TimesInCurrentState -ne 0 ] ; then
        # put umounts here
        /etc/umount /dev/dsk/0s3
        /etc/umount /dev/dsk/0s4
        sync
    fi
    ;;
2)
    if [ $TimesInCurrentState -eq 0 ] ; then
        sync
        sleep 3    # time for the above sync to happen
        /etc/dnld -dd -a 5000000 -f /usr/lib/dnld/imsc -o /dev/im0
    fi
    # put mounts here (/usr, etc.)
    /etc/mount /dev/dsk/0s3 /user
    /etc/mount /dev/dsk/0s4 /lang

```

**Figure 8-7.** Inserting mounts and umounts in `/etc/rc`

---

## Configuring a Second Disk Drive

To configure a second disk drive, perform the same general steps and file modification procedures that you did when you configured the original system disk. These procedures are described in the subsections "Step-by-Step Disk Configuration Overview" and "Step-by-Step Disk Configuration Example" in this chapter.

To run stand-alone **dconfig** on the second drive do the following:

1. At the boot prompt enter:

**dconfig**

2. At the **\$\$dconfig** prompt enter:

**/dev/dsk/1s0**

If you want to run stand-alone **dconfig** on the third physical drive, use **/dev/dsk/2s0**.

3. Follow the displays in Figures 8-2 through 8-5, depending on the type of drive controller you have.

The differences in configuring a second drive (1s[1-15], 2s[1-15], etc.) are listed below:

- You do not need to allocate space on a second disk drive for the root file system or a swap area. Therefore, set rootdev, pipedev, dumpdev, swapdev, swaplo, and nwsap to **0**. When you run stand-alone **dconfig**, enter a **0** to the first six **dconfig** prompts (see Step 12).

**[ NOTE ]** DO NOT change these **dconfig** values for the first drive, **0sx**.

- Since you will not have the root file system or swap space on the second drive, you probably will want to change the settings for **dsk/1sx** when you invoke stand-alone **dconfig**. Using a diagram, plan the division of your second physical disk.

- 
- You must make the nodes for all file systems on the second disk with **mknod**. Refer to Table 8-7 for the correct minor device number. To calculate the minor device number for **1s3** (the third logical disk partition on physical disk 1) add 16 to 3. The resulting value, 19, is used for the minor device parameter in **mknod**. The minor device number for **2s3** is 35 (i.e.  $(2 * 16) + 3$ ). The correct **mknod** syntax to create nodes in **/dev** for **/dev/dsk/1s3** and **/dev/dsk/2s3** is:

```
/etc/mknod /dev/dsk/1s3 b 0 19
/etc/mknod /dev/dsk/2s3 b 0 35
```

- The second disk (**1sx**) can be used for a raw disk (**/dev/rdisk**), such as a data base. The **mknod** syntax to create a node in **/dev** for a raw disk is:

```
/etc/mknod /dev/rdisk/1sx c 3 xx
```

Where: *x* is the logical disk number

**c** stands for character device

**3** is the major character device number for raw disk (**/dev/rdisk**)

*xx* is the minor device number, (see Table 8-7)

- The descriptor that refers to the physical disk (i.e. **1s0**, **2s0**, **3s0**, ... **15s0**) MUST incorporate the first two sectors (sector 0 and sector 1) of the disk (as set with **dconfig**).
- The descriptor which refers to the first logical file system (i.e. **1s1**, **2s1**, **3s1**, ... **15s1**) can (but does not have to) incorporate the first two sectors (sector 0 and sector 1) of the disk. Thus **/dev/dsk/1s1** can be configured to begin at sector 0.
- All other descriptors which refer to the other logical file systems on a disk (**1s2** ... **1s15**, **2s2** .. **2s15**, **3s2** ..**3s15**, etc.) MUST NOT begin at sector 0 or 1. A logical file system MUST begin on an even sector.
- If a logical disk is used as a data base (i.e. it is not used as a file system), it must start after block 1 (sector 0 and sector 1 comprise the first block).

---

**Table 8-8. Software Interleave and Blocks per Cylinder**

| <b>Drive Type</b> | <b>Cont</b> | <b>Sect/<br/>Cyl</b> | <b>cylsize<br/>Blocks/<br/>Cyl</b> | <b>softint<br/>Software<br/>Interleave</b> |
|-------------------|-------------|----------------------|------------------------------------|--------------------------------------------|
| NEC 22Mb 8"       | IMSP        | 102                  | 51                                 | 1                                          |
| NEC 36Mb 8"       | IMSP        | 170                  | 85                                 | 1                                          |
| Fuj. 67Mb 5 ¼"    | MMSP        | 175                  | 88                                 | 9                                          |
| Fuj. 71Mb 8"      | IMSP        | 238                  | 119                                | 1                                          |
| Fuj. 71Mb 8"      | EMSP        | 224                  | 112                                | 7*                                         |
| Fuj. 72Mb 14"     | IMSP        | 170                  | 85                                 | 1                                          |
| Fuj. 72Mb 14"     | EMSP        | 165                  | 82                                 | 7*                                         |
| Fuj. 135Mb 5 ¼"   | MMSP        | 350                  | 175                                | 9                                          |
| Fuj. 142Mb 8"     | AMSP        | 330                  | 165                                | 7*                                         |
| NEC 142Mb 8"      | IMSP        | 272                  | 136                                | 1                                          |
| NEC 142Mb 8"      | EMSP        | 256                  | 128                                | 7*                                         |
| NEC 142Mb 8"      | AMSP        | 248                  | 124                                | 7*                                         |
| Fuj. 145Mb 8"     | IMSP        | 340                  | 170                                | 1                                          |
| Fuj. 145Mb 8"     | EMSP        | 320                  | 160                                | 7*                                         |
| Fuj. 145Mb 8"     | AMSP        | 330                  | 165                                | 7*                                         |
| Fuj. 145Mb 14"    | IMSP        | 340                  | 170                                | 1                                          |
| Fuj. 145Mb 14"    | EMSP        | 330                  | 165                                | 7*                                         |
| NT 270Mb 8"       | IMSP        | 396                  | 198                                | 1                                          |
| Fuj. 285Mb 14"    | IMSP        | 544                  | 272                                | 1                                          |
| Fuj. 285Mb 14"    | EMSP        | 528                  | 264                                | 7*                                         |
| Fuj. 286Mb 8"     | AMSP        | 670                  | 335                                | 7*                                         |
| NEC 286Mb 8"      | EMSP        | 660                  | 330                                | 7*                                         |
| Max. 300Mb 5 ¼"   | MMSP        | 420                  | 210                                | 9                                          |
| Fuj. 310Mb 5 ¼"   | MMSP        | 525                  | 262                                | 9                                          |
| NEC 443Mb 9"      | EMSP        | 1102                 | 551                                | 7*                                         |
| NEC 443Mb 9"      | AMSP        | 1120                 | 560                                | 7*                                         |
| Fuj. 570Mb 14"    | EMSP        | 1056                 | 528                                | 12                                         |
| Fuj. 586Mb 10.5"  | EMSP        | 1320                 | 660                                | 7*                                         |

\* **softint** is optimal for standard Sys5 operation. **softint** can be changed to be optimal for your normal operation.

See the section "Cylinder Size" for equations describing how to compute **mkfs** cylinder size.

# CHANGING ROOT AND SWAP

---

This chapter explains:

- what the swap area is and how it is used.
- how to increase the size of the root file system.
- how to increase or decrease the swap area's size.
- how to move the location of the swap area.

The first part of this chapter explains how to determine the root file system's and swap area's size. Read the beginning of this chapter to determine if your root file system or swap area needs to be modified.

Follow the procedures in this section only if:

- the root file system size needs to be enlarged.
- the swap area needs to be increased, decreased, or the location changed.

**[ NOTE ]** Stand-alone **dconfig** must be run from the boot ( : ) prompt. DO NOT EXECUTE stand-alone **dconfig** when Sys5 is booted and running.



---

## Before You Begin

It cannot be stressed enough that properly configuring your disk is extremely important. Before you begin any procedure described in this chapter, make sure you understand exactly what needs to be done. Time spent planning out how to configure the root file system and the swap area before you begin is well worth time saved when performing the actual tasks. There is nothing worse than finding yourself in the middle of the procedure to increase the size of the root file system and finding out you do not have enough disk space for your file systems.

Make sure you know the total sectors available for your disk (see the chapter *Configuring Disks*) and draw a picture of how you want your disk configured. By doing this, you will ensure that you have enough room on your disk for all file systems.

Take your time and analyze how you want to set up your disk before you begin. And above all, back up everything (ALL file systems) before you begin the procedure. Good luck!

## Root File System Size

If your root file system is not large enough to accommodate your Sys5 software release, see the procedures described later in this chapter to enlarge the size of the root file system.

## An Overview of the Swap Area

Sys5 can read and write data to and from main memory when there are large memory-intensive demands put upon the system. When more memory demands are made upon the system than available memory can store, Sys5 starts to use the disk swap area as a temporary memory storage location. Sys5 copies (swaps) user tasks in and out to effectively manage user demands given the limited memory available.

For example, when a process in main memory has received its share of the job processor's available time, the process is copied to the swap area (swapped out). One or more of the active processes on the disk are copied into main memory (swapped in). Plexus

---

presently uses the `/dev/dsk/0s2` logical disk as the swap area. The swap area is referenced by the `/dev/swap` device node, which should have the same characteristics as the `/dev/dsk/0s2` node.

## Determining the Size of the Swap Area

Plexus systems are shipped with the software installed and the disk mapped. At present, most Plexus Sys5 systems have a factory-set swap area on `/dev/dsk/0s2`. The following factors can help you determine how much swap space is needed:

### Number of Users

As the number of users increases, so does the amount of processing and swapping required by the system. Swap area should be added in proportion to the projected number of concurrent users. Score one point for each user you expect to have on your system at any one time.

### Number of Processes

A system with a few users each running several processes at once is equivalent to a system with several more users. Swap space must be increased according to the projected number of simultaneous processes regardless of the number of people actually logged in. Score one more point for each extra process you anticipate your users will have running in background in addition to interactive tasks.

---

## Types of Processing

Installations for software development, documentation, and scientific applications require more swap area than office environments running primarily accounting and inventory applications unless they are on a database management system. Also, application program size per process dictates the size of the swap area because the entire process is swapped out. Larger programs memory size need a large swap area. Score four points for each database management system, text formatter, other large application program, and/or language compiler that will be in use on a typical day.

## Background Tasks

Different systems (and different system administrators) implement varying amounts of automated background processing. This can take the form of **cron** files, printer schedulers, connecting your system to a telephone line network such as USENET, etc. Score four points for each such task your system will run on a fairly regular basis.

Now total up your points and divide them by four. Round any fractions up to the next whole number. The result is the approximate number of megabytes of swap space you should implement on your system.

The following system messages are symptoms of too small of a swap area:

- No more processes
- Sys5: panic: out of swap space
- Cannot fork

---

## Increasing, Decreasing, or Moving the Swap Area

Plexus configures the `/dev/dsk/0s2` logical disk for the swap area. If the error messages previously listed are displayed, you will need to increase the swap area's size. However, if using the factors discussed in the section "Determining the Size of the Swap Area" you determine that the Plexus default swap area is too large, you might want to decrease the swap area's size. This is especially true with smaller systems.

A general procedure follows. After the general procedure, specific examples are given. It is suggested to first read through the general procedure. Then, refer to the specific examples for greater detail.

A few general considerations are:

- Adjust the `Swplo` sector value accordingly. `Swplo` is the offset for the logical partition identified by `Swapdev`.
- To enlarge the swap area, specify a higher `Nswap` (number of 512-byte sectors for the swap area) beginning at `Swplo`.
- To decrease the swap area, specify a lower `Nswap` (number of 512-byte sectors for the swap area) beginning at `Swplo`.
- To move the swap area, specify the minor device number of the `/dev/dsk/0sxx` of your choice in response to the `Swapdev` prompt. The minor device number is the last number in the device name represented by `xx`. For example, if you are moving the swap space from `0s1` to `0s2`, enter `2` in response to `Swapdev`.
- `Dumpdev` must be the same as `Swapdev`.
- When increasing or decreasing the swap area, make sure the beginning sector address of the next highest `/dev/dsk/0sxx` does not interfere or overlap. This step is extremely important. For example, if you are increasing the size of the root file system, you must move the starting sector for the next logical disk (`/dev/dsk/0s3`) and remake the file system.

---

To increase, decrease, or move the swap area, follow the procedure described below:

1. Determine how large you want the swap area. See the subsection "Determining the Size of the Swap Area" in this chapter.
2. Determine where you want the swap space. The factory set swap area is **/dev/dsk/0s2**. When changing the swap area's size, you might change the logical disk boundaries. If changing file system boundaries, make a chart of the new boundaries, including starting and ending addresses in 512-byte sectors, to use as an aid in the steps that follow. Unless moving the swap area from **/dev/dsk/0s1** to **/dev/dsk/0s2**, to conform with Plexus standards, there is no need to change the location of the swap area.
3. Take the system down to single-user mode (see the chapter *Startup and Shutdown*).
4. Back up the root file system using **/etc/dump**. See the examples later in this chapter.

**[ WARNING ]** Do not use **/etc/dump** if a file system has been reduced in size. **/etc/dump** uses i-node information to decide where to restore the files.

- 
5. Back up ALL other file systems using **cpio**. Mount the first additional file system (such as **/user**) to be backed up; back it up using **cpio**. Lastly, unmount the file system. See the examples later in this chapter for step-by-step examples. Continue this process for ALL file systems (i.e **mount**, **cpio**, **umount**).
  6. Reset the system by pressing RESET but DO NOT BOOT THE SYSTEM. The self-test program activates immediately. As it completes its test, messages appear.
  7. Load the release tape in the tape drive. At the boot prompt (:), invoke stand-alone **dconfig** from tape by entering:

**dconfig**

The system returns

\$\$ dconfig

at the \$\$ dconfig prompt, enter the physical disk name:

**/dev/dsk/0s0**

- 
8. Follow the **dconfig** display presented in the chapter *Configuring Disks*. Keep pressing **<cr>** until you see the message Change the default unix device mapping [y/n]. Answer **y**. The following display is presented. Enter the following responses.

Rootdev     **1**

Pipedev     **1**

Dumpdev     Dumpdev should be the same as Swapdev.

Swapdev     **2** if the swap area is on 0s2; **1** if the swap area is on 0s1.

Swaplo     the starting address in 512-byte sectors of the swap area. For example, if you want your swap area on 0s2 and have a 30Mb root file system on 0s1, enter **0**. If you want your swap space on the end of 0s1 and have a 22Mb root file system on 0s1, enter **44000**.

Nswap     the length in 512-byte sectors of the swap area (beginning at Swaplo). For example, if you are creating a 10Mb swap area, enter **20000**. If you are creating a 34Mb swap area, enter **68000**.

9. After exiting **dconfig**, boot Sys5 by pressing

**<cr>**

at the boot prompt ( : ). This action loads Sys5.

10. Using **mkfs**, re-create any file systems whose boundaries have changed.
11. Restore those file systems from the tape (use **cpio**) and **mount** them.

---

## Increasing the Root File System and the Swap Area

This section provides examples describing in a step-by-step fashion how to increase the size of the root file system and the swap area. Use these examples as models; alter these models as appropriate for your disk configuration.

**[ NOTE ]** Do not enter the information described in these examples verbatim because your disk configuration will probably be different.

### Example 1

It is assumed you have a 142Mb 8" disk. Your root file system on 0s1 occupies 22Mb and your swap area on 0s2 occupies 9Mb. 0s3 and 0s4 have additional file systems of 30Mb and 40Mb, respectively. You want to increase the root file system size to 34Mb (on 0s1) and the swap area to 16Mb (on 0s2).

Table 9-1 illustrates differences between the present disk configuration and what is desired.

**Table 9-1. Example 1 — Disk Configuration**

| <u>Have:</u>            | <u>Want:</u>            |
|-------------------------|-------------------------|
| 22Mb root on 0s1        | 34Mb root on 0s1        |
| 9Mb swap on 0s2         | 16Mb swap on 0s2        |
| 30Mb file system on 0s3 | 30Mb file system on 0s3 |
| 40Mb file system on 0s4 | 40Mb file system on 0s4 |

To increase the size of the root file system to 34Mb and the size of the swap area to 16Mb follow this procedure:

**[ NOTE ]** Take your time and make sure you read through this procedure completely before performing it.



1. Determine how you want to configure your disk. Ensure that you have enough room on your disk taking into account all of the file systems previously set up. See the chapter *Configuring Disks* for the total sectors available (TSA) for your disk. It is recommended to draw a picture or a chart of the way you would like your disk to be configured. The chapter *Configuring Disks* contains examples. The following table illustrates the desired disk configuration.

**Table 9-2. Example 1 — File System Size**

| Device | File System Size | 512-Byte Sectors       |
|--------|------------------|------------------------|
| 0s1    | 34Mb             | 0 to 68000             |
| 0s2    | 16Mb             | 68000 to 100000        |
| 0s3    | 30Mb             | 100000 to 160000       |
| 0s4    | 40Mb             | 160000 to 240000       |
| 0s5    | ??Mb             | 240000 to 272544 (TSA) |

Note that the number, 272544, is the total number of 512-byte sectors available (see Tables 8-2 and 8-3 in the chapter *Disk Configuration*).

This procedure assumes you have enough disk space to keep `/dev/dsk/0s3` 30Mb and `/dev/dsk/0s4` 40Mb. If you do not have enough disk space, you will have to make `/dev/dsk/0s3` and `/dev/dsk/0s4` smaller (see Examples 2 and 3 for more information).

2. Shutdown the system to single-user mode. See the chapter *Startup and Shutdown*.
3. Back up the root file system using `/etc/dump`. Enter:

```
/etc/dump 0uf /dev/xxx/0m /dev/dsk/0s1
xxx is rrm for reel-to-reel tape,
rpt for cartridge tape,
crm for caching reel-to-reel tape (Multibus only),
rft for caching cartridge tape (Multibus only)
```

4. Mount and back up all remaining file systems using `cpio`. DO NOT CONTINUE UNTIL YOU HAVE BACKED UP ALL FILE SYSTEMS. The `cpio` process involves mounting the file system, changing to the mountstub directory,

---

invoking **cpio** to back up the file system, checking your tape with the **-t** option, and unmounting the file system. Enter:

- a. **mount /dev/dsk/0s3 /mntstub3** (i.e. **/user**)
  - b. **cd /mntstub3** (i.e. **/user**)
  - c. **find . -print -cpio /dev/xxx/0m xxx = rpt** for cartridge tape, **rrm** for reel-to-reel tape, **rft** for caching cartridge tape (Multibus only), and **crm** for caching reel-to-reel tape (Multibus only)
  - d. **cpio -itB </dev/xxx/0m**
  - e. **umount /dev/dsk/0s3**
  - f. **mount /dev/dsk/0s4 /mntstub4** (i.e. **/lang**)
  - g. **cd /mntstub4** (i.e. **/lang**)
  - h. **find . -print -cpio /dev/xxx/0m xxx = rpt** for cartridge tape, **rrm** for reel-to-reel tape, **rft** for caching cartridge tape (Multibus only), and **crm** for caching reel-to-reel tape (Multibus only)
  - i. **cpio -itB </dev/xxx/0m**
  - j. **umount /dev/dsk/0s4**
  - k. Continue the **mount**, **cd**, **cpio**, check **cpio** tape, and **umount** procedure for all remaining file systems.
5. Press **RESET**. The system goes through a self-test. At the boot ( : ) prompt, run stand-alone **dconfig**. Insert the release tape in the tape drive. Enter:

**dconfig**

The system responds:

**\$\$ dconfig**

6. At the **dconfig** prompt enter:

**/dev/dsk/0s0**

See the chapter *Disk Configuration* for a sample **dconfig** display. Keep pressing **<cr>** until you see Change the

---

default device mapping. Enter **y**.

7. The following display appears. Ensure that the following are entered as responses.

|               |              |
|---------------|--------------|
| Rootdev [0x0] | <b>1</b>     |
| Pipedev [0x0] | <b>1</b>     |
| Dumpdev [0x0] | <b>2</b>     |
| Swapdev [0x0] | <b>2</b>     |
| Swplo [0x0]   | <b>0</b>     |
| Nswap [0x0]   | <b>32000</b> |

Change the file system disk configuration [y/n] **y**

Note that Rootdev and Pipedev responses (both 1 for 0s1) must be the same. Also, the Dumpdev and Swapdev responses (both 2 for 0s2) must be the same. Swplo, the beginning address in 512-byte sectors of the swap area must be 0. Nswap, the length of the swap area (in 512-byte sectors), is 32000 (16000 X 2).

8. Next the following lines appear. The responses in bold are for the 34Mb root file system and 16Mb swap area. **DO NOT FOLLOW THIS EXAMPLE VERBATIM IF YOU ARE USING DIFFERENT VALUES FOR ROOT AND SWAP.**

---

[sector start,sector count]

```
dsk?s0 [0,0] 0,~
dsk?s1 [0,0] 0,68000
dsk?s2 [0,0] 68000,32000
dsk?s3 [0,0] 100000,60000
dsk?s4 [0,0] 160000,80000
dsk?s5 [0,0] 240000,~
dsk?s6 [0,0] <cr>
dsk?s7 [0,0] <cr>
dsk?s8 [0,0] <cr>
dsk?s9 [0,0] <cr>
dsk?s10 [0,0] <cr>
dsk?s11 [0,0] <cr>
dsk?s12 [0,0] <cr>
dsk?s13 [0,0] <cr>
dsk?s14 [0,0] <cr>
dsk?s15 [0,0] <cr>
```

```
Is the above information correct[y/n] y
Are you sure you want to rewrite block 0?: y
Block 0 of [pd](0,0) initialized successfully!
Exit n (n is 0 if answered y or 1 if answered n)
PLEXUS PRIMARY BOOT REV X.X
:
```

---

For the two values you enter, the first number is the starting sector; the number after the comma is the number of 512-byte sectors comprising the logical file system. It does NOT mean starting sector, ending sector.

For example, 0s0 means the entire disk so the starting sector is 0 and the sector count is ~ which means the entire disk. For 0s1, the root file system starts at sector 0 and the sector count is 68000 (34000 blocks X 2).

The starting sector of 0s2 is 68000 and the sector count is 32000 sectors (16000 blocks X 2). 0s3 begins at 100000 sectors (68000 + 32000) and the sector count is 60000 sectors (30000 X 2). 0s4 begins at 160000 (100000 + 60000) and the sector count is 80000 (40000 X 2). And 0s5 begins at 240000 (160000 + 80000) and the sector count is the end of the disk ~. For the 142Mb 8" disk the total sectors available, noted by ~, is 272544 (see Table 8-2 in the chapter "Disk Configuration").

9. Invoke stand-alone **mkfs** to create a 34Mb root file system for the root file system. Ensure that the release tape in the tape drive. At the boot prompt ( : ) enter:

```
mkfs
```

The system returns:

```
$$ mkfs
```

Enter:

```
/dev/dsk/0s1 34000 softint cylsize
```

For *softint* and *cylsize* information, see Table 9-3.

**Table 9-3.** Software Interleave and Blocks per Cylinder

| Drive Type       | Cont | Sect/<br>Cyl | cylsize<br>Blocks/<br>Cyl | softint<br>Software<br>Interleave |
|------------------|------|--------------|---------------------------|-----------------------------------|
| NEC 22Mb 8"      | IMSP | 102          | 51                        | 1                                 |
| NEC 36Mb 8"      | IMSP | 170          | 85                        | 1                                 |
| Fuj. 67Mb 5 ¼"   | MMSP | 175          | 88                        | 9                                 |
| Fuj. 71Mb 8"     | IMSP | 238          | 119                       | 1                                 |
| Fuj. 71Mb 8"     | EMSP | 224          | 112                       | 7*                                |
| Fuj. 72Mb 14"    | IMSP | 170          | 85                        | 1                                 |
| Fuj. 72Mb 14"    | EMSP | 165          | 82                        | 7*                                |
| Fuj. 145Mb 8"    | AMSP | 330          | 165                       | 7*                                |
| Fuj. 135Mb 5 ¼"  | MMSP | 350          | 175                       | 9                                 |
| NEC 142Mb 8"     | IMSP | 272          | 136                       | 1                                 |
| NEC 142Mb 8"     | EMSP | 256          | 128                       | 7*                                |
| NEC 142Mb 8"     | AMSP | 248          | 124                       | 7*                                |
| Fuj. 145Mb 8"    | IMSP | 340          | 170                       | 1                                 |
| Fuj. 145Mb 8"    | EMSP | 320          | 160                       | 7*                                |
| Fuj. 145Mb 14"   | IMSP | 340          | 170                       | 1                                 |
| Fuj. 145Mb 14"   | EMSP | 330          | 165                       | 7*                                |
| Max. 300Mb 5 ¼"  | MMSP | 420          | 210                       | 9                                 |
| NT 270Mb 8"      | IMSP | 396          | 198                       | 1                                 |
| Fuj. 286Mb 8"    | AMSP | 670          | 335                       | 7*                                |
| Fuj. 285Mb 14"   | IMSP | 544          | 272                       | 1                                 |
| Fuj. 285Mb 14"   | EMSP | 528          | 264                       | 7*                                |
| NEC 286Mb 8"     | EMSP | 660          | 330                       | 7*                                |
| Fuj. 310Mb 5 ¼"  | MMSP | 525          | 262                       | 9                                 |
| NEC 443Mb 9"     | EMSP | 1102         | 551                       | 7*                                |
| NEC 443Mb 9"     | AMSP | 1120         | 560                       | 7*                                |
| Fuj. 570Mb 14"   | EMSP | 1056         | 528                       | 12                                |
| Fuj. 586Mb 10.5" | EMSP | 1320         | 660                       | 7*                                |

\* **softint** is optimal for standard Sys5 operation. **softint** can be changed to be optimal for your normal operation.

**mkfs** cylsize, used when running **mkfs**, can be calculated by using:

$$\text{MCYLSIZE} = (\text{NHEADS} * (\text{NSECTPT} - \text{ALTSECT})) / 2$$

---

See the section "Cylinder Size" in the chapter *Configuring Disks* for additional information. See the subsection "Step-by-Step Disk Configuration Examples" in the chapter *Configuring Disks* for sectors per track and number of heads.

10. In the previous step you erased 0s1 by running **mkfs**. Run **restor** from your Plexus Sys5 release tape to restore your backup **/etc/dump** of **/dev/dsk/0s1**, the minimal file system needed. Follow these steps.

- a. Ensure that the release tape is in the drive.
- b. At the boot prompt ( : ) enter:

```
restor
```

- c. When the following prompt is shown

```
$$ restor
```

take the release tape out of the drive. Put the **/etc/dump** backup tape of your root file system (the backup you made at the beginning of this procedure) into the drive and enter:

```
rf /dev/xxx/0m /dev/dsk/0s1  
xxx is rpt for cartridge tape,  
rrm for reel-to-reel tape,  
rft for caching cartridge tape (Multibus),  
crm for caching reel-to-reel tape (Multibus)
```

The system prompts with the following. Enter what is in bold.

Last chance before scribbling on /dev/dsk/0s1 <er>

- d. The **restor** command reads your **/etc/dump** tape of **/dev/dsk/0s1**. The tape rewinds before the **restor** is complete. After the tape is read to disk, the following messages are displayed:

```
Please wait — building free lists  
EXIT 0
```

---

**[ WARNING ]** Do not press RESET before the system returns Exit 0.

11. At the boot prompt ( : ), press **<cr>**. This loads the kernel **/unix**. When the # prompt appears, you are in single-user mode. For additional information on startup and shutdown, see the chapter *Startup and Shutdown*.
12. Run **/etc/mkfs** for 0s3 and 0s4. Enter:

```
/etc/mkfs /dev/dsk/0s3 30000 softint cylsize
/etc/mkfs /dev/rdisk/0s3 30000 softint cylsize
/etc/mkfs /dev/dsk/0s4 40000 softint cylsize
/etc/mkfs /dev/rdisk/0s5 40000 softint cylsize
```

For *softint* and *cylsize* information see Table 9-3 in this chapter.

13. Mount **/dev/dsk/0s3**. Enter:

```
mount /dev/dsk/0s3 /mntstub3 (i.e. /user)
```

14. Insert the backup tape of **/dev/dsk/0s3** into the drive and change to the mountstub directory. Enter:

```
cd /mntstub3 (i.e. /user)
cpio -idumBv </dev/xxx/0m
xxx is rpt for cartridge tape,
rrm for reel-to-reel tape,
rft for caching cartridge tape (Multibus), and
crm for caching reel-to-reel tape (Multibus)
```

- Mount **/dev/dsk/0s4**. Enter:

```
mount /dev/dsk/0s4 /mntstub4 (i.e. /lang)
```

15. Insert the backup tape of **/dev/dsk/0s4** into the drive and change to the mountstub directory. Enter:



---

**cd /mntstub4** (i.e. */lang*)  
**cpio -idumBv </dev/xxx/0m**  
*xxx* is **rpt** for cartridge tape,  
**rrm** for reel-to-reel tape,  
**rft** for caching cartridge tape (Multibus only), and  
**crm** for caching reel-to-reel tape (Multibus only)

---

## Example 2

This example assumes you have a 142Mb 8" disk. Your root file system on 0s1 occupies 18Mb and your swap area, which also resides on 0s1, occupies 6Mb. 0s2 and 0s3 have additional file systems of 20Mb and 40Mb, respectively. You want to:

- increase the root file system size to 22Mb (on 0s1)
- move the swap area to 0s2 and increase the swap area to 10Mb.

Table 9-4 illustrates differences between the present disk configuration and what is desired.

**Table 9-4. Example 2 — Disk Configuration**

| <u>Have:</u>            | <u>Want:</u>            |
|-------------------------|-------------------------|
| 18Mb root on 0s1        | 22Mb root on 0s1        |
| 6Mb swap on 0s1         | 10Mb swap on 0s2        |
| 20Mb file system on 0s2 | 17Mb file system on 0s3 |
| 40Mb file system on 0s3 | 35Mb file system on 0s4 |

This procedure assumes you do not have enough disk space to keep additional file systems 20Mb and 40Mb in size. Therefore, `/dev/dsk/0s3` and `/dev/dsk/0s4` are 17Mb and 35Mb, respectively. Follow this procedure:

**[ NOTE ]** Take your time and make sure you read through this procedure completely before performing it.

1. Determine how you want to configure your disk. Ensure that you have enough room on your disk taking into account all of the file systems previously set up. See the chapter *Configuring Disks* for the total sectors available (TSA) for your disk. It is recommended to draw a picture or a chart of the way you would like your disk to be configured. The chapter *Configuring Disks* contains examples. The following table illustrates the desired disk configuration.

**Table 9-5. Example 2 — File System Size**

| Device | File System Size | 512-Byte Sectors       |
|--------|------------------|------------------------|
| 0s1    | 22Mb             | 0 to 44000             |
| 0s2    | 10Mb             | 44000 to 64000         |
| 0s3    | 17Mb             | 64000 to 98000         |
| 0s4    | 35Mb             | 98000 to 168000        |
| 0s5    | ??Mb             | 168000 to 272544 (TSA) |

Note that the number, 274544, is the total number of 512-byte sectors available (see Tables 8-2 and 8-3 in the chapter *Disk Configuration*).

2. Shutdown the system to single-user mode. See the chapter *Startup and Shutdown*.
3. Back up the root file system using `/etc/dump`. Enter:

```
/etc/dump u0f /dev/xxx/0m /dev/dsk/0s1  
xxx is rpt for cartridge tape,  
rrm for reel-to-reel tape,  
rft for caching cartridge tape (Multibus),  
crm for caching reel-to-reel tape (Multibus)
```

4. Back up all remaining file systems using `cpio`. DO NOT CONTINUE UNTIL YOU HAVE BACKED UP ALL FILE SYSTEMS. The `cpio` process involves mounting the file system, changing to the `mountstub` directory, invoking `cpio` to backup the file system, checking your `cpio` tape with the `-t` option, and unmounting the file system. Enter:
  - a. `mount /dev/dsk/0s2 /mntstub2` (i.e. `/user`)
  - b. `cd /mntstub2` (i.e. `/user`)
  - c. `find . -print -cpio /dev/xxx/0m`  
`xxx` = `rpt` for cartridge tape,  
`rrm` for reel-to-reel tape,  
`rft` for caching cartridge tape (Multibus only), and  
`crm` for caching reel-to-reel tape (Multibus only)
  - d. `cpio -itB </dev/xxx/0m`  
`xxx` is `rrm` for reel-to-reel tape,  
`rpt` for cartridge tape,

- 
- rft** for caching cartridge tape (Multibus only), and  
**crm** for caching reel-to-reel tape (Multibus only)
- e. **umount /dev/dsk/0s2**
  - f. **mount /dev/dsk/0s3 /mntstub3** (i.e. /user)
  - g. **cd /mntstub3** (i.e. /user)
  - h. **find . -print -cpio /dev/xxx/0m**  
*xxx* = **rpt** for cartridge tape,  
**rrm** for reel-to-reel tape,  
**crm** for caching reel-to-reel tape (Multibus only), and  
**rft** for caching cartridge tape (Multibus only)
  - i. **cpio -itB </dev/xxx/0m**  
*xxx* is **rrm** for reel-to-reel tape,  
**rpt** for cartridge tape,  
**crm** for caching reel-to-reel tape (Multibus only), and  
**rft** for caching cartridge tape (Multibus only)
  - j. **umount /dev/dsk/0s3**
  - k. Continue the **mount**, **cd**, **cpio**, check **cpio** tape, and  
**umount** procedure for all remaining file systems.
5. Press RESET. The system goes through a self-test. At the boot ( : ) prompt, run stand-alone **dconfig**. Insert the release tape in the tape drive. Enter:

**dconfig**

The system responds:

\$\$ dconfig

6. At the dconfig prompt enter:

**/dev/dsk/0s0**

See the chapter *Disk Configuration* for a sample **dconfig** display. Keep pressing **<cr>** until you see Change the default device mapping. Enter **y**.

7. The following display appears. Ensure that the following are entered as responses.

---

|               |              |
|---------------|--------------|
| Rootdev [0x0] | <b>1</b>     |
| Pipedev [0x0] | <b>1</b>     |
| Dumpdev [0x0] | <b>2</b>     |
| Swapdev [0x0] | <b>2</b>     |
| Swplo [0x0]   | <b>0</b>     |
| Nswap [0x0]   | <b>20000</b> |

Change the file system disk configuration [y/n] y

Note that Rootdev and Pipedev responses (both 1 for 0s1) must be the same. Also, the Dumpdev and Swapdev responses (both 2 for 0s2) must be the same. Swplo, the beginning address in 512-byte sectors of the swap area must be 0. Nswap, the length of the swap area (in 512-byte sectors) is 20000 (10000 blocks X 2).

- Next, the following lines appear. The responses in bold are for the 22Mb root file system and 10Mb swap area. **DO NOT FOLLOW THIS EXAMPLE VERBATIM IF YOU ARE USING DIFFERENT VALUES FOR ROOT AND SWAP.**

[sector start,sector count]

|               |                    |
|---------------|--------------------|
| dsk?s0 [0,0]  | <b>0,~</b>         |
| dsk?s1 [0,0]  | <b>0,44000</b>     |
| dsk?s2 [0,0]  | <b>44000,20000</b> |
| dsk?s3 [0,0]  | <b>64000,34000</b> |
| dsk?s4 [0,0]  | <b>98000,70000</b> |
| dsk?s5 [0,0]  | <b>168000,~</b>    |
| dsk?s6 [0,0]  | <b>&lt;cr&gt;</b>  |
| dsk?s7 [0,0]  | <b>&lt;cr&gt;</b>  |
| dsk?s8 [0,0]  | <b>&lt;cr&gt;</b>  |
| dsk?s9 [0,0]  | <b>&lt;cr&gt;</b>  |
| dsk?s10 [0,0] | <b>&lt;cr&gt;</b>  |
| dsk?s11 [0,0] | <b>&lt;cr&gt;</b>  |
| dsk?s12 [0,0] | <b>&lt;cr&gt;</b>  |
| dsk?s13 [0,0] | <b>&lt;cr&gt;</b>  |
| dsk?s14 [0,0] | <b>&lt;cr&gt;</b>  |
| dsk?s15 [0,0] | <b>&lt;cr&gt;</b>  |

---

```
Is the above information correct[y/n]  y
Are you sure you want to rewrite block 0?:  y
Block 0 of [pd](0,0) initialized successfully!
Exit n (n is 0 if answered y or 1 if answered n)
PLEXUS PRIMARY BOOT REV X.X
```

For the two values you enter, the first number is the starting sector; the number after the comma is the number of 512-byte sectors comprising the logical file system. For example, 0s0 means the entire disk so the starting sector is 0 and the sector count is ~ which means the entire disk. For 0s1, the root file system starts at sector 0 and the sector count is 44000 (22000 X 2).

The starting sector of 0s2 is 44000 and the sector count is 20000 sectors (10000 X 2). 0s3 begins at 64000 sectors (44000 + 20000) and the sector count is 34000 sectors (17000 X 2). 0s4 begins at 98000 (64000 + 34000) and the sector count is 70000 (35000 X 2). And 0s5 begins at 168000 (98000 + 70000) and the sector count is the end of the disk ~. For the 142Mb 8" disk the total sectors available is 272544. The end of the disk is noted is ~.

9. Invoke stand-alone **mkfs** to create a 22Mb root file system. Ensure that the Plexus Sys5 release tape is in the tape drive. At the boot prompt ( :) enter:

```
mkfs
```

The system returns:

```
$$ mkfs
```

Enter:

```
/dev/dsk/0s1 22000 softint cylsize
```

For *softint* and *cylsize* information, see Table 9-3 in the section "Example 1" earlier in this chapter.

10. In the previous step you erased 0s1 by running **mkfs**. Run **restor** from your Plexus Sys5 release tape to restore your **/etc/dump** of **/dev/dsk/0s1**, the minimal file system needed.

---

Follow these steps.

- a. Ensure that the release tape is in the drive.
- b. At the boot prompt ( : ) enter:

**restor**

- c. When the following prompt is shown

**\$\$ restor**

take the release tape out of the drive. Put the **/etc/dump** backup tape of your root file system (the backup you made at the beginning of this procedure) into the drive and enter:

**rf /dev/xxx/0m /dev/dsk/0s1**

**xxx** is **rpt** for cartridge tape,

**rrm** for reel-to-reel tape,

**crm** for caching reel-to-reel tape (Multibus only),

**rft** for caching cartridge tape (Multibus only)

The system prompts the following. Enter what is in bold.

Last chance before scribbling on /dev/dsk/0s1 **<cr>**

- d. The **restor** command reads your **/etc/dump** of **/dev/dsk/0s1**. The tape rewinds before the **restor** is complete. The following messages are displayed:

Please wait — building free lists

EXIT 0

**[ WARNING ]** Do not press RESET before the system returns EXIT 0.

11. At the boot prompt ( : ), press **<cr>**. This loads the kernel **/unix**. When the # prompt appears, you are in single-user mode. For additional information on startup and shutdown, see the chapter *Startup and Shutdown*.

- 
12. After ensuring you have enough space for the 35Mb file system make the node for 0s4. Enter:

```
mknod /dev/dsk/0s4 b 0 4
```

13. Remove **/dev/swap**. Remake **/dev/swap** on 0s2, and set the read/write/execute permissions. Enter:

```
rm /dev/swap
mknod /dev/swap b 0 2
chmod 440 /dev/swap
```

Run **/etc/mkfs** for **/dev/dsk/0s3** and **/dev/dsk/0s4**. Enter:

```
/etc/mkfs /dev/dsk/0s3 17000 softint cylsize
/etc/mkfs /dev/rdisk/0s3 17000 softint cylsize
/etc/mkfs /dev/dsk/0s4 35000 softint cylsize
/etc/mkfs /dev/rdisk/0s4 3500 softint cylsize
```

For *softint* and *cylsize* information see Table 9-3 in this chapter.

14. Mount 0s3. Enter:

```
mount /dev/dsk/0s3 /mntstub3 (i.e. /user)
```

15. Insert the backup tape of **/dev/dsk/0s2** into the drive. Since you moved the swap area to **/dev/dsk/0s2**, now your file system is on **/dev/dsk/0s3**. Change to the mountstub directory and **cpio** in your files. Enter:

```
cd mntstub3 (i.e. /user)
cpio -idumBv </dev/xxx/0m
xxx is rpt for cartridge tape,
rrm for reel-to-reel tape,
crm for caching reel-to-reel tape (Multibus), and
rft for caching cartridge tape (Multibus)
```



---

Mount 0s4. Enter:

```
mount /dev/dsk/0s4 /mntstub4 (i.e. /lang)
```

16. Check the amount of disk free space. Enter:

```
df -t
```

which returns the disk free space. You want to make sure that your file system that will be on 0s4 (previously 0s3) will fit in 35Mb (previously 40Mb).

17. Insert the backup tape of 0s3 (because you have moved the swap area to 0s2 now your file system is on 0s4) into the drive. Change to the mountstub directory and **cpio** in your files. Enter:

```
cd mntstub4 (i.e. /lang)
cpio -idumBv </dev/xxx/0m
xxx is rpt for cartridge tape,
rrm for reel-to-reel tape,
crm for caching reel-to-reel tape (Multibus),
rft for caching cartridge tape (Multibus)
```

---

### Example 3

This procedure assumes you are happy with the size of your root file system but your swap space is too small. Your swap area which is on 0s2 is 6Mb. You want to increase the size of the swap area to 14Mb. Your root file system on 0s1 is 30Mb. The file system on 0s3 is 30Mb and the file system on 0s4 is 40Mb. Because you do not have enough disk space, you need to reduce the size of `/dev/dsk/0s3` to 26Mb and `/dev/dsk/0s4` to 36Mb. You have a 285Mb 14" disk (EMSP controller).

Table 9-6 illustrates differences between the present disk configuration and what is desired.

**Table 9-6. Example 3 — Disk Configuration**

| <u>Have:</u>            | <u>Want:</u>            |
|-------------------------|-------------------------|
| 30Mb root on 0s1        | 30Mb root on 0s1        |
| 6Mb swap on 0s2         | 14Mb swap on 0s2        |
| 30Mb file system on 0s3 | 26Mb file system on 0s3 |
| 40Mb file system on 0s4 | 36Mb file system on 0s4 |

Follow this procedure:

**[ NOTE ]** Take your time and make sure you read through this procedure completely before performing it.

1. Determine how you want to configure your disk. Ensure that you have enough room on your disk taking into account all of the file systems previously set up. See the chapter *Configuring Disks* for the total sectors available (TSA) for your disk. It is recommended to draw a picture or a chart of the way you would like your disk to be configured. The chapter *Configuring Disks* contains examples. The following table illustrates the desired disk configuration.

**Table 9-7. Example 3 — File System Size**

| <b>Device</b> | <b>File System Size</b> | <b>512-Byte Sectors</b> |
|---------------|-------------------------|-------------------------|
| 0s1           | 30Mb                    | 0 to 60000              |
| 0s2           | 14Mb                    | 60000 to 88000          |
| 0s3           | 26Mb                    | 88000 to 140000         |
| 0s4           | 36Mb                    | 140000 to 212000        |
| 0s5           | ??Mb                    | 212000 to 539616        |

Note that 539616 is the total number of 512-byte sectors available (see Tables 8-2 and 8-3 in the chapter *Disk Configuration*).

2. Shutdown the system to single-user mode. See the chapter *Startup and Shutdown*.
3. Back up the root file system using `/etc/dump`. Enter:

```
/etc/dump 0 /dev/xxx/0m /dev/dsk/0s1  
xxx is rpt for cartridge tape,  
rrm for reel-to-reel tape,  
crm for caching reel-to-reel tape (Multibus),  
rft for caching cartridge tape (Multibus)
```

4. Back up all remaining file systems using **cpio**. DO NOT CONTINUE UNTIL YOU HAVE BACKED UP ALL FILE SYSTEMS. The **cpio** process involves mounting the file system, changing to the mountstub directory, invoking **cpio** to backup the file system, checking your **cpio** tape with the **-t** option, and unmounting the file system. Enter:

- a. `mount /dev/dsk/0s3 /mntstub3` (i.e. `/user`)
- b. `cd /mntstub3` (i.e. `/user`)
- c. `find . -print -cpio /dev/xxx/0m`  
*xxx* = **rpt** for cartridge tape,  
**rrm** for reel-to-reel tape,  
**crm** for caching reel-to-reel tape (Multibus only), and  
**rft** for caching cartridge tape (Multibus only)
- d. `cpio -itB </dev/xxx/0m`  
*xxx* is **rpt** for cartridge tape,  
**rrm** for reel-to-reel tape,

---

**crm** for caching reel-to-reel tape (Multibus), and  
**rft** for caching cartridge tape (Multibus)

- e. **umount /dev/dsk/0s3**
  - f. **mount /dev/dsk/0s4 /mntstub4** (i.e. **/lang**)
  - g. **cd /mntstub4** (i.e. **/lang**)
  - h. **find . -print -cpio /dev/xxx/0m xxx = rpt** for cartridge tape, **rrm** for reel-to-reel tape, **crm** for caching reel-to-reel tape (Multibus only), and **rft** for caching cartridge tape (Multibus only)
  - i. **cpio -itB </dev/xxx/0m xxx** is **rpt** for cartridge tape, **rrm** for reel-to-reel tape, **crm** for caching reel-to-reel tape (Multibus only), and **rft** for caching cartridge tape (Multibus only)
  - j. **umount /dev/dsk/0s4**
  - k. Continue the **mount**, **cd**, **cpio**, and **umount** procedure for all remaining file systems.
5. Reset the system by pressing RESET but DO NOT BOOT THE SYSTEM. The self-test program activates immediately. As it completes its tests, messages appear.
6. Insert the release tape in the tape drive. At the boot prompt (:), invoke stand-alone **dconfig** from tape. Enter:

**dconfig**

The system returns:

\$\$ **dconfig**

At the \$\$ **dconfig** prompt, enter the logical disk name:

**/dev/dsk/0s0**

See the chapter *Disk Configuration* for a sample **dconfig** display. Keep pressing **<cr>** until you see Change the default device mapping. Enter **y**.

- 
7. The following display appears. Ensure that the following are entered as responses.

|               |              |
|---------------|--------------|
| Rootdev [0x0] | <b>1</b>     |
| Pipedev [0x0] | <b>1</b>     |
| Dumpdev [0x0] | <b>2</b>     |
| Swapdev [0x0] | <b>2</b>     |
| Swplo [0x0]   | <b>0</b>     |
| Nswap [0x0]   | <b>28000</b> |

Change the file system disk configuration [y/n]y

Note that Rootdev and Pipedev responses (both 1 for 0s1) must be the same. Also, the Dumpdev and Swapdev responses (both 2 for 0s2) must be the same. Swplo, the beginning address in 512-byte sectors of the swap area must be 0. Nswap, the length of the swap area (in 512-byte sectors), is 28000 (14000 blocks X 2).

8. Next the following lines appear. The responses in bold are for the 30Mb root file system, the 14Mb swap area, a 26Mb 0s3 and a 36Mb 0s4. **DO NOT FOLLOW THIS EXAMPLE VERBATIM IF YOU ARE USING DIFFERENT VALUES FOR ROOT AND SWAP.**

---

[sector start,sector count]

```
dsk?s0 [0,0] 0,~
dsk?s1 [0,0] 0,60000
dsk?s2 [0,0] 60000,28000
dsk?s3 [0,0] 88000,52000
dsk?s4 [0,0] 140000,72000
dsk?s5 [0,0] 212000,~
dsk?s6 [0,0] <cr>
dsk?s7 [0,0] <cr>
dsk?s8 [0,0] <cr>
dsk?s9 [0,0] <cr>
dsk?s10 [0,0] <cr>
dsk?s11 [0,0] <cr>
dsk?s12 [0,0] <cr>
dsk?s13 [0,0] <cr>
dsk?s14 [0,0] <cr>
dsk?s15 [0,0] <cr>
```

```
Is the above information correct[y/n]  y
Are you sure you want to rewrite block 0?:  y
Block 0 of [pd](0,0) initialized successfully!
Exit n (n is 0 if answered y or 1 if answered n)
PLEXUS PRIMARY BOOT REV X.X
```

:

For the two values you enter, the first number is the starting sector; the number after the comma is the number of 512-byte sectors comprising the logical file system. For example, 0s0 means the entire disk so the starting sector is 0 and the sector count is ~ which means the entire disk. For 0s1, the root file system starts at sector 0 and the sector count is 60000 (30000 blocks X 2).

The starting sector of 0s2 is 60000 and the sector count is 28000 sectors (14000 blocks X 2). 0s3 begins at 88000 sectors (60000 + 28000) and the sector count is 52000 sectors (26000 blocks X 2). 0s4 begins at 140000 (88000 + 40000) and the sector count is 72000 (36000 blocks X 2). And 0s5 begins at 212000 (140000 + 72000) and the sector count is the end of the disk ~. For the 285Mb 14" disk the total

---

sectors available is 539616. The end of the disk is noted by

9. At the boot prompt ( : ), press **<cr>**. This loads the kernel, **/unix**. When the # prompt appears, you are in single-user mode. For additional information on startup and shutdown, see the chapter *Startup and Shutdown*.
10. Run **/etc/mkfs** for 0s3 and 0s4. Enter:

```
/etc/mkfs /dev/dsk/0s3 26000 softint cylsize
/etc/mkfs /dev/rdisk/0s3 26000 softint cylsize
/etc/mkfs /dev/dsk/0s4 36000 softint cylsize
/etc/mkfs /dev/rdisk/0s4 36000 softint cylsize
```

For *softint* and *cylsize* information see Table 9-3 in this chapter.

11. Mount 0s3. Enter:

```
mount /dev/dsk/0s3 /mntstub3 (i.e. /user)
```

12. Insert the backup tape of 0s3 in the tape drive. Change to the mountstub directory and **cpio** in your files. Enter:

```
cd mntstub3 (i.e. /user)
cpio -idumBv </dev/xxx/0m
xxx is rpt for cartridge tape,
rrm for reel-to-reel tape,
crm for caching reel-to-reel tape (Multibus only),
rft for caching cartridge tape (Multibus only)
```

Mount 0s4. Enter:

```
mount /dev/dsk/0s4 /mntstub4 (i.e. /lang)
```

13. Insert the backup tape of 0s4 into the tape drive. Enter:

```
cd mntstub4 (i.e. /lang)
cpio -idumBv </dev/xxx/0m
xxx is rpt for cartridge tape,
rrm for reel-to-reel tape,
crm for caching reel-to-reel tape (Multibus only), and
rft for caching cartridge tape (Multibus only)
```

# CREATING USER ACCOUNTS

---

This chapter provides instructions for implementing the login id and home directory required for each new user. The following procedures are discussed:

- creating login directories
- creating `/etc/passwd` and `/etc/group` entries
- creating `.profile` (Bourne shell) or `.cshrc` (C shell)
- changing individual and group ownership
- setting the path
- exporting `TERM` and `PATH`
- setting terminal type

A troubleshooting guide is also presented. To use this chapter effectively, you must have a user file system (such as `/user`) set up. This is where user accounts reside.

**[ NOTE ]** Do not use `/usr` for user accounts. If you have not set up a user directory, see the chapter *Configuring Disks*.



---

This chapter describes how to set up user accounts in one of many possible ways. For additional information, see the *Sys5 UNIX User's Reference Manual* and the *Sys5 UNIX Programmer's Reference Manual*. Table 10-1 provides a command summary.

**Table 10-1. Command Summary**

| <b>Subject</b> | <b>Command/Title</b>                                                                                   |
|----------------|--------------------------------------------------------------------------------------------------------|
| password       | <b>passwd</b> (1)                                                                                      |
| group          | <b>group</b> (4)<br><b>chgrp</b> (see <b>chown</b> (1))                                                |
| profile        | <b>.profile</b> (4) (Bourne shell)<br>OR<br><b>.cshrc</b> (4) (C shell)<br><b>.login</b> (1) (C shell) |
| terminal type  | <b>term</b> (4), <b>terminfo</b> (4)                                                                   |
| shell          | <b>sh</b> (1), <b>csh</b> (1)                                                                          |

---

## Procedure to Add a User

To add a new user, you must do the following:

- Create a login directory (home directory) under each user's name.
- Assign that directory to the new user.
- Insert this name and certain parameters into `/etc/passwd` and optionally `/etc/group`.
- If your users will be using the Bourne shell (`/bin/sh`), create a general `.profile` file and copy it into each user's directory (or modify `/etc/profile`).
- If your users will be using the C shell (`/usr/plx/csh`), create a general `.login` and `.cshrc` file and copy these files into each user's directory (or modify `/etc/cshprofile`).
- Optimize the `.profile` file (Bourne shell) OR `.cshrc` and `.login` files (C shell) for each user's account.
- Set the proper ownership and protections for the user's directory and any files.

## Creating login Directories

Following is the procedure for creating login directories:

1. Login as root.
2. Create directories named after each person or account you wish to add to your system with the following command:

```
mkdir /user/username
```

where:

*username* is the name of the person or account created.  
*username* must be 7 characters or less (alphanumeric only for best results).

For example, to create an account for someone named John in the `/user` directory, enter:

---

**mkdir /user/john**

3. Repeat the previous command until you have created directories for everyone being added who will have login privileges to the system on a regular basis. These home directories give new users a place to work.

### Creating /etc/passwd Entries

The following procedure describes how to configure the **/etc/passwd** file, which defines user identity.

1. Be sure you are logged in as root or become super-user.
2. Make sure you are familiar with the **/etc/passwd** file and the information fields required for each entry. These are described in **passwd(1)** and **passwd(4)** in the *Sys5 UNIX User's Reference Manual* and the *Sys5 UNIX Programmer's Reference Manual*.
3. Copy the **passwd** file to a backup version to guard against accidents:

```
cp /etc/passwd /etc/passwd.bu
```

4. Edit the **/etc/passwd** file. Create entries in **/etc/passwd** for each user name represented by a directory.
5. Create and fill in the appropriate fields. The colons ( **:** ) are field separators on each entry line. Below are two different examples:

```
john::103:1:John Niswonger:/user/john:/usr/plx/csh  
tj::104:1:Theresa Jackson:/user/tj:/bin/sh
```

An explanation of the examples above follows:

```
lognm:nopswd:uid:gid:cmt:homelogdir:logprog
```

where:

```
lognm = login name
```

---

*nopswd* = no password  
*uid* = user id  
*gid* = group id  
*cmt* = comment field  
*homelogdir* = home login directory  
*logprog* = login program or shell

From these entries we see that John's numerical user id is 103 and tj's is 104; we know their real names to avoid confusion during turnover (comment field); John's home login directory is */user/john* and tj's is */user/tj*; and that John uses the C shell while tj uses the standard, or Bourne shell. Make sure the user id is unique and the login name is eight characters or less.

**[ NOTE ]** When assigning user id numbers, start at 101 and progress from there. By convention (not required), user id numbers 1 through 100 are reserved for system use.

You do not enter anything in the password field itself (second from the left). That space gets filled with an encrypted password automatically when the individual user invokes the **passwd(1)** program as you did for root. Because */etc/passwd* is a read only file, you must force writing of the file. For **ed** use **w! q**. The new user can change the password when (s)he logs in later.

If, for security reasons, you want passwords for all users before they even log in, invoke **passwd** by entering

**passwd loginname**

to set up passwords for each user.

---

## Creating /etc/group Entries

1. If you plan to use group security as well, look at the `/etc/group` file. Enter:

```
cat /etc/group
```

Below is a partial listing of the `/etc/group` file. The colons ( : ) are field separators on each entry line.

```
root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
mail::6:root
rje::8:rje,shqer
guest::100:guest
```

The information fields are:

- first field is the group name
- second field (blank) is the encrypted password (not used)
- third field is the numerical group id
- fourth field is a comma-separated list of all users allowed in the group

2. The other group is a default group. This group is a catch-all group. If you want your users to be in the other group, edit the `/etc/passwd` file and enter **1** for the group id (GID) number field in the individual's `passwd` entry.
3. If you want your users to be members of a new group, first add a line to the `/etc/group` file following the format previously described. Use a group number and name not previously used. Limitations to the `/etc/group` file follow. For example, if you want users divided into three groups clerks, staff, and director, add these three groups to the `/etc/group` file. Your file would like something like this:

---

```
root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
mail::6:root
rje::8:rje,shqer
daemon::12:root,daemon
clerks::101:gjones,msmith,bwhite
staff::102:kblack,jdoe,bhenry
director::103:blogan,gjones
```

Note that the user `gjones` is in two groups. There is not a limit as to the number of users in each group. Also, there is not a limit on the number of groups that a particular user can be a part of. Additional information describing `group(4)` is in the *Sys5 UNIX Programmer's Reference Manual*.

4. Next, edit the `/etc/passwd` file to define the group id number field correctly. This will establish which group the user becomes affiliated with when the user logs in. If the user is in more than one group, choose the group (s)he will want to be part of most often and assign that group id in `/etc/passwd`. Note that this user can change his effective group id when he is logged in by executing `newgrp(1)`.

#### Limitations on the `/etc/group` File

Some limitations imposed on the `/etc/group` file are:

- If more than one line is required to enter a large number of members, the beginning of the second line must have the first three (3) fields of the group entry repeated.
- Group names should not be longer than seven (7) characters and should contain alphanumeric characters only.
- Do not make your users members of the default system groups `bin`, `sys`, `adm`, etc. The other group is provided as a general catch-all user group, if you do not wish to create groups of your own.

---

## Creating `.profile` or `.cshrc`

The files `/etc/profile` (Bourne shell) and `/etc/csh/profile` (C shell) are the master login profiles that will set up parameters for all users running shells. The files `.profile` and `.login`, in the users' home directories, are used to further customize a user's environment beyond which the master `/etc/profile` file establishes. Creating and configuring `.profile` and/or `.cshrc` files for the system's user accounts saves much time and confusion when the users log onto the system and try to use it.

Without a `.profile` (Bourne shell) or a `.cshrc` (C shell) file, users would have to specify the paths to each command they invoked. Also, some Berkeley-originated utilities would not work properly. Follow the procedure below to set up `.profile` or `.cshrc` files.

1. Create a model `.profile` for accounts logging into the `/bin/sh` (Bourne shell) environment and a model `.cshrc` for accounts logging into the `/usr/plx/csh` (C shell) environment. Below are sample `.profile` and `.cshrc` files:

### Sample `.profile`

```
stty erase "^h" kill "^x" echoe
TERM=vt100
PATH=:/user/$LOGNAME:/usr/plx:/bin:/usr/bin:/usr/lib
EXINIT='set aw wm=15 number redraw'
MAIL=/usr/mail/$LOGNAME
TZ=PST8PDT
export TERM PATH EXINIT MAIL TZ
```

To set up the prompt and other shell variables, see `sh(1)` in the *Sys5 UNIX User's Reference Manual*. `EXINIT` is defined in the chapter VI in the *Sys5 UNIX User Guide*. For additional information on `TZ`, see `environ(5)` in the *Sys5 UNIX Programmer's Reference Manual*.

---

### Sample .cshrc

```
setenv EXINIT 'set aw wm=15 number redraw'  
setenv TERM vt100  
setenv PATH /usr/plx:/bin:/usr/bin:/usr/lib:/user/john:  
setenv MAIL /usr/mail/$LOGNAME  
setenv TZ PST8PDT  
setenv SHELL /bin/csh  
stty erase '^h' kill '^x' ixon echoe  
set history=20  
set prompt=' % '  
alias a alias  
a h history  
a lo logout
```

Notice that the **.cshrc** contains more parameters than **.profile** because the Bourne shell does not support aliases or the history function while the C shell features both.

2. Modify **/etc/profile** (Bourne shell) or **/etc/cshprofile** (C shell) and add definitions of environment variables or commands that apply to all users. Modifying **/etc/profile** or **/etc/cshprofile** saves space and setup time. If you use the Bourne shell, you might not need the **.profile** file for all users.
3. Copy **.profile** into each new user directory that will use the Bourne shell as its environment:

```
cp .profile userdir
```

Repeat this command for each appropriate account.

4. Make the files executable and owned by the individual owners:

```
chmod 755 userdir/.profile  
chown loginname userdir/.profile
```
5. Copy your model **.cshrc** into each new user directory that will use the C shell as its environment:

```
cp .cshrc userdir
```



---

Repeat this command for each new user directory.

6. Make **.profile** or **.cshrc** executable and owned by the individual users. Enter:

```
chmod 755 userdir/.profile OR
chmod 755 userdir/.cshrc
chown loginname userdir/.profile OR
chown loginname userdir/.cshrc
```

7. Edit each user's **.profile** or **.cshrc** file and specify the individual's account name on the line specifying the MAIL environment (or specify \$LOGNAME). **.profile** (Bourne shell) or **.cshrc** (C shell) is executed at the time the shell is forked when the user logs in.
8. (Optional) For accounts running C shell, you might want to create **.login** files as well as **.cshrc** files. The difference is that the **.login** file sets parameters for the login shell only, whereas the **.cshrc** sets parameters for all forked shells as well. The **.login** file is not necessary for system or user operation, however. Refer to the chapter, *CSH* in the *Sys5 UNIX User's Guide* for a more detailed description of the function of the **.login** file and suggestions for appropriate parameters.

## Setting the Path (Bourne Shell)

The variable, **PATH**, is defined on the third line in the **.profile** example. When a user enters a command that is not part of the standard shell, Sys5 searches in all the directories in the command path until it finds the command (unless the user specifies exactly where the command is by using an absolute pathname for the command).

The format of the command path line for the Bourne shell only is:

```
PATH=:/xxxx/xxxx:/yyyy/yyyy:/zzzz/zzzz
```

The colon (:) is a field separator that separates the different path names.

---

To make the command (PATH) available to other sub-shells, add the command line below to the **.profile** file after the line for the command path:

```
export PATH
```

A typical system has commands in **/usr/plx**, **/bin**, **/usr/bin**, **/usr/lib**, (which contains lp admin files) and **/homedir**. Commands and shell procedures created especially for **/user/john** usually go in **/user/john/bin** (this is optional). Before John can execute the commands in these locations, the pathnames must be specified in his **.profiles** path assignment.

Other directories can be added to the path list. For example, if John wanted to use the commands in Sara's bin, he could add **/user/sara/bin** to his path list. However, there is a cost in system resources; Sys5 scans every location in the path list whenever a command is entered. To promote efficiency, keep the path list as short as possible and order the directories so that the most often used appear earlier in the list.

To initialize the changes to the current environment after modifying the user's **.profile** file, the the user enters:

```
. /homedir/.profile
```

For example, a user, John, who wants to initialize changes to his **.profile** file enters:

```
. /user/john/.profile
```

---

## Setting the Path (C Shell)

To set PATH for the C shell, add the following line to the `.cshrc` or `.login` file. Exporting variables is not necessary when using `setenv` in the C shell.

```
setenv PATH /usr/plx:/bin:/usr/bin:/usr/lib:/user/john:
```

To initialize the changes to the current environment after modifying the user's `.cshrc` or `.login` file. Enter:

```
source /homedir/filename
```

For example, a user, John, who wants to initialize changes to his `.cshrc` file enters:

```
source /user/john/.cshrc
```

If you need to set the terminal type, see the section "Setting the Terminal Type" in this chapter.

## Changing Individual and Group Ownerships

A basic Sys5 principle is that anything you create is yours. If you log in as root and create directories and files, all those directories and files belong to root. Since root set up login directories for all the user accounts, root owns them. Now you (as root) must relinquish the ownership of the relevant directories and files to the accounts designed to use them.

**[ NOTE ]** You MUST perform the procedures in this section or the user will not be able to log in.

Change ownerships of login directories and special files with the following procedures:

1. Since only the owner of a file can change the permissions or ownership of a file, you must be logged in as root to change the ownership of a file or directory you do not own.

- 
2. Change your working directory to the **/user** directory.
  3. Change the ownership of each user's home directory to the owner's login name. Changing the ownership lets each user own his login directory. This lets the user control who has access to his files. Enter:

```
chown loginname homedir
```

For example, to give John ownership of his own login directory (named john) enter:

```
chown john john
```

4. Change the group (**chgrp**) of the user's home directory to the group the user belongs to as defined in **/etc/passwd**:

```
chgrp groupname homedir
```

For example, if John's group id or group name in **/etc/passwd** (field number 4) is 1 (other), enter:

```
chgrp other john
```

5. If you want each account to own his or her **.profile** file (Bourne shell) or **.cshrc** file and **.login** files (C shell), change ownership of each of these files with **chown** and the standard syntax:

```
chown username /homedir/.profile
```

```
chown username /homedir/.login
```

```
chown username /homedir/.cshrc
```

6. If your terminals are properly connected and configured, your system is now ready to accept logins from everyone assigned to it. Be sure to test each login to make sure you have not skipped a step or mistyped information in these files before telling users their accounts are set up.

---

## Changing Group Affiliations

This subsection explains how users assigned to multiple groups can change their group affiliation. Users who are assigned to only one group cannot take on other group identities. Group assignments must be entered into the `/etc/group` file by the system administrator by entering the user names in the appropriate group definition lines. A user, such as Jones, who is a member of two groups assumes a default affiliation with the `clerks` group because of the entry in `/etc/passwd`. That user must intentionally change his or her group affiliation to have access to files owned by the other group, which are protected from the rest of the world. Use the `newgrp` command to change group affiliation. Enter:

```
newgrp director
```

This command first reads the `/etc/group` file to see if his user name is listed as a valid member of the target group. If so, he will then become a member of the group `director`. Otherwise, the affiliation change will not work and he will be told `Sorry`. If the transition to becoming `director` works, he no longer has special access rights with his former group's files (`clerks`) since he is now affiliated with another group. He must change back to `clerks` with the `newgrp` command. Any files `gJones` creates after executing the `newgroup` command will be owned by the group `director`.

A common mistake is in assigning users to a default user group, such as the `other` group, and forgetting to add the username to the `/etc/group` entry for `other`. If this user changes his/her affiliation to another group and then tries to change back to `other`, the `newgrp` command will fail because (s)he is not listed as a member. It is important to monitor that every user you add to the system (by editing the `/etc/passwd` file) is also added to one or more groups in `/etc/group` file. You will not have to edit the `/etc/group` file if the default group for the user in `/etc/passwd` is `other` (`group id = 1`).

The user can change the group ownership associated with a file or directory owned by him or her. A normal listing of a directory (with the `ls -l` command) will show the listing, the group

---

ownername appears right after the individual ownername.

To change the group ownership of a file, use the **chgrp** command. Its usage is similar to changing the individual ownership of the file with **chown**. If gJones has a file called **work** that has the group ownership of director, and he wants his clerks to be able to access it instead, he enters:

```
chgrp clerks work
```

The super-user can change group ownership and individual ownership regardless of who owned the file or directory initially, but all other users can only change the group and individual ownership of the files they own outright.

## Setting Terminal Type

Because of the differences between terminals, some of the UC Berkeley-originated programs (such as **more**) need to know what type of terminal they will interface. This information is contained in the directory hierarchy, **/usr/lib/terminfo/n/name** (*name* is the identifier for a specific terminal and *n* is the first letter of that name).

The **terminfo** directory contains many common terminal types and provides some code to deal with their various differences. To make this information available to the programs that require it, add the following line to the **.profile** (Bourne shell) file:

```
TERM=name  
export TERM
```

For the C shell, enter the following information in the **.cshrc** file.

```
setenv TERM name
```

where:

*name* is one of the terminal names provided in **/usr/lib/terminfo/n/name** (some terminal types have many names in **/usr/lib/terminfo/n/name**).

For additional information on **terminfo(4)**, see the chapter *Curses and Terminfo Package* in the *Sys5 UNIX Programmer's Guide* and **terminfo(4)** in the *Sys5 UNIX Programmer's Reference*

---

*Manual.*

Some applications running under Sys5 might still refer to Sys3 **/etc/termcap**. If so, the value of TERM must be defined in that file. Installation instructions for those applications should explain which files are references. Make sure the terminals you will be using are defined appropriately.

## **Troubleshooting Guide**

Once you have created a login directory for a user, if the user is unable to log in to that account, use the following troubleshooting list to try to track down the problem. Table 10-2 presents a troubleshooting guide.

---

**Table 10-2.** Troubleshooting Guide

| <b>Situation</b>                                   | <b>Possible Problem</b>                                                                   | <b>Corrective Action</b>                                                                                                                                                                                                                                                            |
|----------------------------------------------------|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| no login:<br>prompt appears<br>on user's terminal. | 1. System in single-user state (state s).                                                 | 1a. From console, enter <b>who -r</b> to check if the system is running in multi-user or single-user mode. Bring system to multi-user mode (state 2).                                                                                                                               |
|                                                    | 2. Terminal or cable not properly set up. Check <b>/etc/inittab</b> entry (action = off). | 2a. Check tty connection.<br>2b. Find tty number.                                                                                                                                                                                                                                   |
|                                                    | 3. <b>inittab</b> not set up properly.                                                    | 3a. Check <b>/etc/inittab</b> entry (action = off).                                                                                                                                                                                                                                 |
|                                                    | 4. System down or hung process.                                                           | 4a. Check operability of console and all other terminals. If all are stuck, shut-down the system then press RESET and reboot the system.<br><br>4b. Determine tty number of user. Run <b>ps -aef</b> to check for processes running on that tty; kill them in descending PID order. |



---

**Table 10-2. Troubleshooting Guide (cont)**

| <b>Situation</b>                                                     | <b>Possible Problem</b>                                                                               | <b>Corrective Action</b>                                                                                                  |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| User gets login incorrect: message after many tries.                 | 1. User forgot password or password incorrect.                                                        | 1a. Login as root.<br>Enter:<br><b>passwd <i>username</i></b><br>and redefine user's password.                            |
|                                                                      | 2. <b>passwd</b> file entry for user is incorrect.                                                    | 2a. Edit <b>/etc/passwd</b> and carefully check users entry (or delete it). Verify that user is using correct login name. |
| User gets past login: and passwd: but gets the message No directory. | 1. 6th field in user's <b>passwd</b> entry for login directory path wrong or a non-existent directory | 1a. Edit <b>/etc/passwd</b> . Check for proper number of fields (especially the sixth one).                               |
|                                                                      |                                                                                                       | 1b. Check if the user's directory exists according to the sixth field in <b>passwd</b> .                                  |
|                                                                      |                                                                                                       | 1c. Check ownership/ protections of home directory and user directory permissions to allow access.                        |
|                                                                      | 2. The alternate file system containing user directory is not mounted.                                | 2a. <b>Is</b> the mountstub to see if file system is mounted.                                                             |

---

**Table 10-2.** Troubleshooting Guide (cont)

| <b>Situation</b> | <b>Possible Problem</b>                                            | <b>Corrective Action</b>                                                                                                                                    |
|------------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| No shell.        | 1. <b>passwd</b> entry is wrong.                                   | 1a. Edit <b>/etc/passwd</b> . Check for proper number of fields for user entries. The last field must reference an existing program.                        |
|                  | 2. Target program missing or protected from the user.              | 2a. Check to see that program exists and is executable by the user.                                                                                         |
|                  | 3. Program called from <b>passwd</b> file is a shell script.       | 3a. Change this. Only compiled programs can be called from users <b>passwd</b> entry.<br><br>3b. Permissions on <b>/bin/sh</b> or <b>/bin/su</b> are wrong. |
|                  | 4. Permissions on <b>/bin/sh</b> or <b>bin/su</b> are not correct. | 4a. <b>/bin/sh</b> should be <b>rwxrwxr-x (775)</b> ; <b>/bin/su</b> should be <b>rwsr-xr-x (4755)</b> .                                                    |



# BACKUP PROCEDURES

---

This chapter, which discusses how to make tape backups for your Plexus system, explains how to use the following programs.

- **/etc/dump** and stand-alone **restor**
- **volcopy**
- **fbackup** (not available on VMEbus-based systems)
- **dd**
- **cpio**
- **tar**
- **finc** (and **frec** to recover files)

---

## Types of Backups

Table 11-1 lists the tape utilities by function. The backup programs can be divided into two categories:

- those that recognize discrete files and copy them as such
- those that create a tape image of a logical disk

**Table 11-1. Types of Backup Utilities**

| <b>Utilities that Recognize Discrete Files and Copy Them to Tape</b> | <b>Utilities that Create a Tape Image of a Logical Disk</b> |
|----------------------------------------------------------------------|-------------------------------------------------------------|
| <b>find(1M)</b>                                                      | <b>volcopy(1M)</b>                                          |
| <b>cpio(1)</b>                                                       | <b>fbackup</b>                                              |
| <b>/etc/dump(1M)</b>                                                 | <b>dd(1)</b>                                                |
| <b>tar(1)</b>                                                        |                                                             |
| <b>dd(1) (one file only)</b>                                         |                                                             |

Files on a **find** tape can be recovered with the **freec** command. Use **/etc/restor** to recover a backup made with **/etc/dump**. Use stand-alone **restor** if restoring a backup of the root file system.

One reason there are so many archiving programs is that each was created to fill a different need. The following paragraphs describe each archiving program and its special characteristics.

**tar** was created specifically for tape archiving. **tar** permits limited manipulation of tape block size. It allows you to specify exactly which files you wish to move. Invoking options to **tar** allows you to retrieve files selectively.

**cpio** was created as an I/O copying utility. It can redirect the output to tape or any other device. **cpio** permits limited manipulation of tape block size. It allows you to specify exactly which files you wish to move. Invoking options to **cpio** allows you to retrieve files selectively.

**dd** performs EBCDIC-ASCII conversion and allows very flexible specification of tape block size. **dd** allows you to specify exactly which files you wish to move. Invoking options to **dd** allows you to retrieve files selectively.

---

**finc** was created specifically to perform incremental backups. **finc** was designed to replace **/etc/dump** and is used to backup entire file systems. **finc** does not specifically provide an option to archive everything on a file system because it does this by default. Instead, it provides many options to specify the controlling increment for the files that get backed up. For example, you can specify a backup of every file on a file system that was accessed within the last week or of every file that was changed within the last day. **finc**'s counterpart, **frec**, allows you to restore single files from tape.

**/etc/dump** is used to back up whole file systems. **/etc/dump** either puts the entire file system on tape or puts on tape all the components of the file system that have been altered since the last dump. The first type of dump is called a full or level 0 dump. The second dump, which archives only components that have changed, is called an incremental or level 9 dump. Up to nine levels of incremental dumps (1-9) can be specified.

**/etc/restor** is a counterpart of **/etc/dump**. **/etc/restor** allows you to restore file systems and single files from your **dump** backup tape. If restoring the root file system from your **dump** backup tape, you can use the stand-alone version of **restor**. Stand-alone **restor** is used at the boot prompt ( : ), i.e. before **/unix** is loaded.

**volcopy** is used to back up whole file systems (not individual files). **volcopy** makes a literal copy of the file system and uses a blocksize matched to the device. The **volcopy** files must include labels via **labelit**.

**fbackup** is used to back up whole file systems (not individual files). **fbackup** is not available on VMEbus-based systems (P/90s and P/95s). It is a fast, stand-alone Plexus program that copies from disk to tape (or tape to disk) in streaming mode, blocking files at 32 disk sectors per tape block.

---

The following table summarizes the differences among these programs:

**Table 11-2. Sys5 Programs That Write to Tape**

| <b>Program</b>   | <b>Usage Notes</b>                                                                                                                     | <b>Restore Using</b>                                                                                          | <b>Remarks</b>                                                        |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------|
| <b>/etc/dump</b> | For use on file systems. Allows complete and incremental dumps.                                                                        | Stand-alone <b>restor</b> or <b>/etc/restor</b> . Use stand-alone <b>restor</b> only on <b>/dev/dsk/0s1</b> . | Multi-volume capability.                                              |
| <b>cpio</b>      | For use on files. Writes to standard output. Use with <b>find</b> .                                                                    | <b>cpio</b>                                                                                                   | Multi-volume capability.                                              |
| <b>tar</b>       | For use on files, and recursively on directories.                                                                                      | <b>tar</b>                                                                                                    | Does not handle multiple tape volumes.                                |
| <b>dd</b>        | For use on files and file systems. Does data conversion. Good for I/O on raw devices. Same format as <b>fbackup</b> if <b>bs=16b</b> . | <b>dd</b>                                                                                                     | Does not handle multiple tape volumes. Also available in stand-alone. |

---

**Table 11-2.** Sys5 Programs That Write to Tape (cont)

| <b>Program</b> | <b>Usage Notes</b>                                                  | <b>Restore</b>              | <b>Remarks</b>                                                                         |
|----------------|---------------------------------------------------------------------|-----------------------------|----------------------------------------------------------------------------------------|
| <b>fbackup</b> | For use on file systems. Fast. Same format as <b>dd</b> .           | <b>fbackup</b> or <b>dd</b> | Stand-alone only. Can't restore single files. Not on VMEbus systems (P/90s and P/95s). |
| <b>volcopy</b> | For use on file systems. Uses labels. Good for disk to disk backup. | <b>volcopy</b>              | Can't restore single files.                                                            |
| <b>finc</b>    | For use on file systems. Allows complete & incremental backups.     | <b>frec</b>                 | Does not handle multiple tape volumes.                                                 |

See the relevant pages of the *Sys5 UNIX User's Reference Manual* and the *Sys5 UNIX Administrator's Reference Manual* for further information.



---

## When to Make Backups

You must make some sort of daily tape backup, even if you record only the changes made in the file system(s) from the previous day. This could take the form of an incremental `/etc/dump` or selective archiving of active user files with `tar` or `cpio`. You should also ensure that the system receives a complete backup at least once per week. See the chapters *Configuring Disks* and *Changing Root and Swap* for step-by-step instructions describing how to back up the root file system and any additional file systems.

In addition, some installations use `cpio` to backup and restore all the file systems on a regular basis (monthly, quarterly, or annually) depending on how much frequent disk I/O degrades overall performance) to correct possibly fragmentation of the file system. As a disk drive is accessed, it allocates the next free block on the list. Free blocks are not contiguous after a while because of deleted files. While the system can still find the files, they become more spread out, slowing disk access time. This is referred to as fragmentation.

---

## Utilities to Backup File Systems

This section discusses how to use the following utilities to back up file systems:

- stand-alone **fbackup**
- **volcopy**
- **/etc/dump** (and its counterpart **/etc/restor**)
- **finc**, **frec**, and **ff**

[ **NOTE** ] Before performing tape backups, check the file systems with **fsck**. See the chapter *Stand-alone Programs* for more information.

### Using **fbackup**

This program copies from disk to tape or from tape to disk while using the tape in streaming mode. It is not available on VMEbus-based systems (P/90s and P/95s). It blocks files at 32 512-byte sectors per record.

For example, to copy a file system that is 40000 sectors long (512-byte sectors), starting at offset 100000 (i.e. **/dev/dsk/0s3**) on disk drive 0, onto the tape in cartridge tape unit 0, respond to the boot prompt as indicated in **bold**:

1. Ensure you have at least one blank tape.
2. You must be at the boot prompt ( : ) to run **fbackup**. See the chapter *Startup and Shutdown* to get to the boot ( : ) prompt.
3. Insert the tape into the tape drive. Ensure that the tape is properly loaded (see the chapter *Getting Started* for instructions on how to operate the tape drive).
4. At the boot ( : ) prompt enter:

---

## **/stand/fbackup**

The program responds:

**\$\$ fbackup**

Backup to tape or Restore from tape? [br]: **b**

Disk unit? [dk] [xy(0-3,0)] or [pd(0-3,0)]: **pd(0,0)†**

Tape unit? [rm(0-3,0)] or [pt(0,0)]: **pt(0,0)‡**

Starting disk sector number for backup/restore? [0,xxxxx]: **100000**

Number of sectors in backup/restore? [1,xxxxx]: **40000**

Disk reads complete!

Backup to tape completed successfully

Exit 0

PLEXUS PRIMARY BOOT REV X.X COMPLETE

:

5. You can restore your **fbackup** tape in one of two ways: with **fbackup** or with **dd**. To restore your **fbackup** tape with **fbackup**, use the **r** response shown in the previous example (i.e. Backup to tape or Restore from tape). With **fbackup**, you can restore anywhere by the values you use for the start sector on disk and the number of sectors. You can also restore your tape using the Sys5 UNIX command **dd(1)** with **bs=16b**.

## **Using volcopy**

**volcopy**, a Sys5 utility, copies file systems with label checking. **volcopy** makes a literal copy of the file system using a blocksize matched to the device. That is, it copies the entire file system to tape similar to stand-alone **fbackup**. With **volcopy** you do not have to specify sector addresses (just the name of the file system). **volcopy** determines where to start.

---

† Enter **dk** for an AMSP drive, **xy** for an EMSP drive, or **pd** for an IMSP drive.

‡ Enter **rm** for a reel-to-reel tape drive or **pt** for a cartridge tape drive.

---

**volcopy** saves on tape every file in the file system. Because it performs no selection, it is very fast. You can also use **volcopy** to bring back a file system from tape to disk or to perform disk to disk backups.

To copy a file system from disk to tape perform the procedure described below:

1. Ensure that you have at least one blank tape.
2. Bring the system down to single-user mode. See the chapter *Startup and Shutdown*.
3. Insert the tape into the tape drive. Ensure that the tape is properly loaded (see the chapter *Getting Started* for instructions on how to operate the tape drive).
4. Execute the **volcopy** command. It is assumed that file systems have been labeled using **labelit** as described in the chapter *Disk Configuration*. Enter:

```
/etc/volcopy [options] fname special1 volname1 special2 volname2
```

The options are **-a** and **-s**:

**-s** (default) invoke the **DEL if wrong** verification sequence

**-a** invoke a verification sequence requiring a positive operator response instead of the standard 10-second delay before the copy is made.

*fname* is the name assigned to the logical disk with the **labelit** command. *fname* is recorded in the last 12 characters of the superblock (**char fname[6], volname[6]**) To determine *fname* at the # prompt enter:

```
/etc/labelit /dev/dsk/xsxx (x is the major device number and xx is the minor device number). The system returns fname and volname1.
```

*srcdev* is the disk device name where the copy of the file system is being extracted (such as **/dev/dsk/0s3**). This is the file system you are backing up.

---

*volname1* is the volume from which the copy of the file system is being extracted. *volname1* is recorded in the last 12 characters of the superblock (**char fsname[6], volname[6]**) Use **/etc/labelit** as described under *fsname* to determine *volname1*.

*destdev* is the target device name (the device name the file system is being copied to such as **/dev/rrm/0m**). *destdev* is the device (tape or logical disk) that will contain the backup.

*volname2* is the volume name of the target device. That is, the volname that was assigned with **/etc/labelit** to the device that will contain the backup. To determine the volume name, invoke **/etc/labelit** (see the instructions under *fsname*).

For additional information on **volcopy(1M)**, see the *Sys5 UNIX Administrator's Reference Manual*.

## Using dump for Backups

This section describes how to create backup tapes using **/etc/dump(1M)**. The **/etc/dump** command provides a system for making routine archive tapes which can then be used to recover entire file systems if data on the disk is damaged or lost.

### Scheduling Backups

A rigidly enforced schedule of backups provides the best insurance against lost data. **BACKING UP THE SYSTEM SHOULD NEVER BE IGNORED.**

There are two levels of **dump**: full and incremental. A full **dump** copies everything; an incremental **dump** copies only those files changed since the last **dump** was taken.

A typical schedule requires:

- a. Taking an incremental dump after every working day.

- 
- b. Taking a full dump after work on Fridays.

Quarterly, you might also want to take a full **dump** and then **restor** the file systems to disk to maintain system disk I/O performance.

The tapes made for archival purposes are stored and used in rotation so that the oldest tape is always the next used. Incremental **dump** tapes are saved at least a week and full **dumps** are saved for at least a month.

**dump** is actually a Sys3 command. Plexus has kept it on Sys5 systems because of user demand. Be careful not to confuse **/etc/dump(1M)**, the utility which dumps file systems to tape, with the Sys5 command **dump(1)**, which is used to dump selected parts of an object file. The commands **/etc/dump(1M)** and **dumpdir(1M)** are described in the *Sys5 UNIX Administrator's Reference Manual*.

An alternate dump schedule, which works equally well, is suggested in the description of the **/etc/dump** command.

Installations with more than one logical file system **MUST** take a separate dump for each logical file system.

See the section "Automating your Backup Procedures" in this chapter for a script which dumps the root file system using **/etc/dump** and two additional file systems using **cpio** to tape.

#### **Procedure to Take a Full Dump**

Use the following procedure to take a full **dump** of a file system:

1. Obtain a tape and insert the tape into the tape drive.
2. Ensure that all users are off the system. If necessary, take the system to single-user mode (state s) as described in the chapter, *Startup and Shutdown*.
3. At the # prompt enter:

```
sync  
sync
```

- 
4. Check all file systems with **fsck** to be sure there are no errors. See the section "Repairing File Systems" in the chapter *Crisis Recovery* for information on **fsck**.

5. Enter:

```
/etc/dump 0uf[s] /dev/xxx/0m [XXXX] /dev/dsk/Xsxx
```

where:

- 0** is the dump level
- u** writes the date of the beginning of the dump on the file **/etc/ddate** if the **dump** completes successfully.
- f** places the **dump** on the following argument, specifying the tape drive.
- s** is the size of the dump tape in feet.
- xxx** is the device name: **rpt** for cartridge tape, **rrm** for reel-to-reel tape, **rft** for caching cartridge tape (Multibus only), and **crm** for caching reel-to-reel tape (Multibus only).
- XXXX** is the size of the tape in feet. Enter **1000** for 20Mb cartridge tape drives, **2000** for 45Mb cartridge tape drives, and **2500** for 60Mb cartridge tape drives.
- X** is the physical device number (0 - 16)
- xx** is the logical device number (0 - 15)

6. Wait for the **dump** to finish.
7. If there is too much data for one tape, the system prompts you to load a second tape.
8. When the **dump** is finished, the screen displays information about the **dump**. Copy this information onto the tape label(s).
9. Remove the tape from the drive, mark the label, and store the tape(s).

---

### Procedure to Make Incremental Dumps

1. Follow steps 1-4 in the procedure to make full dumps.
2. After verifying the file system with **fsck**, enter:

```
/etc/dump [1-9]uf[s] /dev/xxx/0m [XXXX] /dev/dsk/Xsxx
```

where:

**1-9** is the dump level.

**u** All files modified since the last date stored in the file **/etc/ddate** for the same file system at lesser levels will be dumped.

See the full **dump** example for explanations of all other values.

3. Wait for the **dump** to finish.
4. If there is too much data for one tape, the system prompts you to load a second tape.
5. When the **dump** is finished, the screen displays information about the **dump**. Copy this information onto the tape label.
6. Remove the tape from the drive, mark the label, and store the tape(s).



---

## Using `restor`

The program `/etc/restor(1M)` is the counterpart of `/etc/dump`. It puts the contents of **dump** tapes back onto the disk. See the section "Using `dump` for Backups" in this chapter for information on full and incremental dumps. Stand-alone **restore** is also available for restoring the root file system.

A few things to remember about using **restor**:

- The file system being restored must exist on the disk.
- Like `/etc/dump`, **restor** works in terms of file systems. It is hard to retrieve single files from a dump tape for two reasons: **restor** uses the i-node number to identify the file you need and **restor** is an older program. However, it is possible to restore single files using the `x` option with `/etc/restore`. It is suggested to use the copy utilities mentioned in this chapter for archiving single files, such as **cpio**.
- **restor** replaces the previous contents of the file system.
- A backup tape prepared with `/etc/dump` must be loaded into a file system that is as large or larger than the file system from which the dump was taken.

**[ WARNING ]** The Sys5 version of **restor** cannot read multiple volume **dumps** made with the Sys3 version of **dump**. If you have multiple volume **dumps** of a Sys3 file system, use the stand-alone **restor** on your old Sys3 release tape to load the **dump** onto your new Sys5 file system. Then use the Sys5 version of `/etc/dump` to make a new backup.

### Using `/etc/restor`

Follow the procedure described below to restore a file system from your **dump** backup tape.

- 
1. Ensure you have at least one blank tape.
  2. Insert the tape into the tape drive. Ensure that the tape is properly loaded (see the chapter *Getting Started* for instructions on how to operate the tape drive).
  3. At the # prompt enter:

```
restor rf /dev/xxx/0m /dev/dsk/0s1
```

where: **xxx** is **rpt** for cartridge tape,  
**rrm** for reel-to-reel tape,  
**crm** for caching reel-to-reel tape (Multibus only), and  
**rft** for caching cartridge tape (Multibus only)

The program returns the following. Press <cr> to restore the file system.

```
Last chance before scribbling on /dev/dsk/0s1 <cr>  
End of tape  
Building free list
```

### Using Stand-alone restor

Use stand-alone **restor** only when restoring the root file system, **/dev/dsk/0s1**. Stand-alone **restor** works the same as **/etc/restor** with the following differences.

- The **x** key is not supported in stand-alone **restor**. This means that you cannot restore individual files with stand-alone **restor**.
- An argument of the form "+<file no>," where <file no> is a positive integer, can (but does not have to) follow the last argument. This directs stand-alone **restor** to skip over <file no> number of files on tape before doing the restore from tape.

To restore the root file system from the release tape, follow the procedure described below:

1. Ensure you have at least one blank tape.
2. You must be at the boot prompt ( : ) to run stand-alone **restor**. See the chapter *Startup and Shutdown* to get to the boot ( : ) prompt.

---

3. Insert the tape into the tape drive. Ensure that the tape is properly loaded (see the chapter *Getting Started* for instructions on how to operate the tape drive).

4. At the boot ( : ) prompt enter:

```
/stand/restor
```

5. The program returns:

```
$$ restor
```

6. At the \$\$ restor prompt enter:

```
rf /dev/xxx/0m /dev/dsk/0s1 +20
```

where: **xxx** is **rpt** for cartridge tape,  
**rrm** for reel-to-reel tape,  
**crm** for caching reel-to-reel tape (Multibus only), and  
**rft** for caching cartridge tape (Multibus only)

The program returns the following. Press <cr> to begin the restore.

```
Spacing forward 20 files on tape  
Last chance before scribbling on /dev/dsk/0s1 <cr>  
End of tape  
Building free list
```

### Restoring an Individual File

With the **x** option of **/etc/restor**, you can restore an individual file from your **dump** tape. Follow the procedure described below. This procedure describes how to restore a file named **testfile**.

1. Insert the correct tape into the tape drive.
2. If you are logged in as root change to the root ( / ) directory. If you are logged in as an ordinary user, make sure you change to a directory you can access.
3. If you are using cartridge tape, ensure that your cartridge tape is rewound. If the tape is not rewound enter:

```
/usr/plx/tape rew
```

---

To determine the correct pathname to restore, use the **dumpdir** command. You can list the contents of the file system in a file. In the example below the contents of the file system will be redirected to a file called **hold**. If you know the file name, you can use **grep** to search for the file name. Enter:

```
/usr/plx/dumpdir f /dev/xxx/0m > hold
```

OR

```
/usr/plx/dumpdir f /dev/xxx/0m | grep test
```

where:

*xxx* is **rrm** for reel-to-reel tape,

**rpt** for cartridge tape,

**crm** for caching reel-to-reel tape (Multibus only), or

**rft** for caching cartridge tape (Multibus only)

4. If you entered the first command, use **cat hold** or **more hold** to view the file named **hold** and determine the correct path name.

If you entered the last command the system returns the i-node number and name of the file such as:

```
345 /mike/testfile
```

5. Change to the directory where you want to restore, rewind the tape, then enter the **restor** command. Enter:

```
cd dirname
```

```
/usr/plx/tape rew
```

```
/etc/restor xf /dev/xxx/0m /mike/testfile
```

*xxx* is **rrm** for reel-to-reel tape,

**rpt** for cartridge tape,

**crm** for caching reel-to-reel tape (Multibus only),

**rft** for caching cartridge tape (Multibus only)

The program returns information and prompts you for the volume number. A volume is a complete dump cycle. Enter what is in boldface type.

---

/mike/testfile i-node 345

Mount desired tape volume: Specify vol #: 1  
extract file 345

You might have more than one volume (**dump** cycle) on the **dump** tape. If you do and **restor** can not find the file name on volume 1, the program reprompts you for the volume number. Enter **2**. **dump** keeps prompting you for the correct volume number until it finds the file on the **dump** tape. Continue incrementing the volume number by 1.

6. **/etc/restor** restores the file with the i-node number as the file name. For example, the i-node number of **testfile** is 345. When the file is restored from tape to disk, the file name will be 345.
7. Verify that file 345 was extracted from tape and copied to your current working directory. Enter:

**ls**

The system returns a listing of your current working directory. Use the **mv** command to change the file name from 345 to a descriptive file name:

**mv 345 testfile**

---

## Utilities to Back Up Individual Files

This section discusses how to use the following utilities to back up individual files or perform incremental backups:

- `cpio`
- `tar`
- `finc` (`frec` to restore files)
- `dd`

### Using `cpio`

This subsection provides a beginning lesson in the use of `cpio(1)`. `cpio` is a powerful command with many options. It is explained completely in the *Sys5 UNIX User's Reference Manual*. To make backups of file systems (except for `root`), it is suggested to use `cpio`.

#### Copying Files to Tape With `cpio`

The following procedure describes how to copy a directory to tape.

1. Insert a tape into the tape drive. Ensure it is properly loaded. See the chapter *Getting Started* for additional information on loading tapes.
2. Change to the correct directory. In this example, the directory `lang` is being copied to tape.
3. `cd lang`
4. At the prompt enter:

```
find . -print -cpio /dev/xxx/0m
```

`cpio` finds the current directory and copies all files and sub-directories to tape.

5. After `cpio` has copied the directory `lang` to tape, ensure that `lang` was copied to tape by entering:

```
cpio -itvB </dev/xxx/0m
```

---

where:

- i** Means copy in. This options extracts files from tape.
- t** prints a table of contents of the input. No files are created.
- v** means verbose. This option causes a list of file names to be printed. When used with the **t** option, the table of contents looks like the output of an **ls -l** command.
- B** is the blocking factor. The input/output is blocked at 5120 blocks to the record.
- xxx** is **rpt** for cartridge tape,  
**rrm** for reel-to-reel tape,  
**crm** for caching cartridge tape (Multibus only), and  
**rfp** for caching reel-to-reel tape (Multibus only)

**cpio** returns a list of files and subdirectories copied to tape.

### Copying Files From Tape to Disk With **cpio**

To restore (copy in) the contents of the tape to your current directory follow this procedure.

1. Change to the directory you want information copied to.

**cd** *dirname*

2. To copy a specific file from tape to your current directory, first list the contents of the tape. Enter:

**cpio -itvB </dev/xxx/0m**

**cpio** returns a list of files and subdirectories copied to tape.

3. Copy the file listed on the list to disk. Enter:

**cpio -ivB filename </dev/xxx/0m**

4. If you want to copy the tape's contents to your current working directory, enter:

**cpio -ivB </dev/xxx/0m**

The above action copies the contents of tape to your current working directory on disk.

---

## Automating your Backup Procedures

Because it is important to make regular backups, Plexus has included a script which backs up the root file system (0s1) using `/etc/dump`, the `/user` file system (0s3), and the `/sccs` file system (0s7).

**[ NOTE ]** Do not enter this script verbatim. It is included as a guide for you to follow. Edit this script as necessary to accommodate your backup needs.

```
#This script dumps (incrementally) the / (root),
#/user and /sccs filesystems. The /user and /sccs
#filesystems are backed up using cpio.
#root ( / ) is being dumped with /etc/dump.
#   root /           ----- /dev/dsk/0s1
#   /user           ----- /dev/dsk/0s3
#   /sccs           ----- /dev/dsk/0s7
#
#
# set -x (uncomment for debug mode)
# /user and /sccs will be dumped using cpio.
for dir in /user /sccs
do
    sync;sync
    cd $dir
    find . -mtime -2 -print -cpio /dev/rrm/0mn
    date > LASTINCRBKUP #file will be used later
    chmod 400 LASTINCRBKUP
done
/usr/plx/tape rew
# rewind 9-track tape
for dir in /user /sccs
do
    rm MESSAGES
    echo >> MESSAGES
    echo "\nSYSTEM ID :   uname -a \n" >> MESSAGES
```



---

```
date >> MESSAGES
echo "\nList $dir files backed up with cpio :\n" >> MESSAGES
cpio -itvB < /dev/rrm/0mn >> MESSAGES
/usr/plx/tape srcheof 1 # position for reading next file
pr MESSAGES | lp
done
/usr/plx/tape rew
/usr/plx/tape srcheof 2
# / (root) will be dumped with /etc/dump
# root will be the third tapefile
# Incremental dump of root (/dev/dsk/0s1)
cd /
sync;sync
/etc/dump u9f /dev/rrm/0mn /dev/dsk/0s1 | mail root
rm ROOTMSGs
echo >> ROOTMSGs
echo "\nSYSTEM ID :  uname -a \n" >> ROOTMSGs
date >> ROOTMSGs
echo "\nList pwd (root) files backed up with dump\n" >> ROOTMSGs
/usr/plx/tape rew
/usr/plx/tape srcheof 2
dumpdir >> ROOTMSGs
pr ROOTMSGs | lp
/usr/plx/tape unload
```

**Figure 11-1.** Sample Backup Script

---

## Using tar to archive

**tar**(1) is a Sys3 program. However, AT&T has included **tar** in the Sys5 software release. **tar** is a tape archive program and might not be included in future releases. Therefore, you are encouraged to use **cpio**.

This section presents a brief overview describing how to copy files or directories to tape and how to extract files or directories from tape using **tar**. Complete instructions describing how to use **tar** are in the *Sys5 UNIX User's Reference Manual*.

To copy the entire contents of a directory called **demo** to tape follow the procedure described below:

1. Load the tape in the tape drive (see the chapter *Getting Started* for instructions on how to use the tape drive).
2. Change to the directory ABOVE where you want to copy from. For example, if you want to copy files and directories in **/user/tom/demo** to tape, **cd** to **/user/tom**. Enter:
3. Enter:

```
tar cvf /dev/xxx/0m demo
```

where:

- c** means create new tape. Writing begins at the beginning of the tape instead of after the last file.
  - v** means verbose. Normally, **tar** does its work silently. The **v** (verbose) option causes **tar** to type the name of each file it treats, preceded by the function letter.
  - f** causes **tar** to use the next argument as the name of the archive (the tape device name).
- xxx** is **rpt** for cartridge tape,  
**rrm** for reel-to-reel tape,  
**crm** for caching reel-to-reel tape (Multibus only), and  
**rft** for caching cartridge tape (Multibus only)

All files and directories under **demo** will automatically be archived to tape.

- 
4. To check that the file was written to tape enter:

```
tar -tvf /dev/xxx/0m
```

where:

- t** lists the file names on the tape.
- v** verbose; lists more information such as permissions, size, and ownership (like the `ls -l` command).
- f** specifies the file used for output (i.e. the tape drive device name)

The system returns a list of the tape's contents.

To extract the directory **demo** from tape, follow the procedure described below:

1. Change to the directory you want to copy to. Enter:

```
cd dirname
```

2. Load the tape in the tape drive.
3. Obtain a listing of the tape's contents, enter:

```
tar tvf /dev/xxx/0m
```

where: *xxx* is **rpt** for cartridge tape,  
**rrm** for reel-to-reel tape,  
**crm** for caching reel-to-reel tape (Multibus only), and  
**rft** for caching cartridge tape (Multibus only)

The system returns the tape's contents.

4. To ensure you do not overwrite an existing **demo** directory, enter:

```
ls -la
```

5. Verify that the file name **demo** is not in your current working directory. Enter:

```
tar xvf /dev/xxx/0m demo
```

where:

- 
- x** means extract files.
  - v** means verbose
  - f** specifies the file used for output (the tape device name)
  - xxx** is **rpt** for cartridge tape, **rrm** for reel-to-reel tape, **rft** for caching cartridge tape (Multibus only), and **crm** for caching reel-to-reel tape (Multibus only)

The **demo** directory is extracted from tape and is created UNDER your present directory. All directories will be created on disk as needed.

## Using find, findc, and frec

**findc** (fast incremental backup) and **frec** (file recovery program) are counterparts. At present, it is recommended to use other copy utilities instead of **findc** and **frec**. **findc** is used to make partial or incremental backups. **findc** selectively copies the input file system to the output raw tape.

Before you make a backup with **findc** you must label the tape with the **labelit** command. The **labelit** syntax follows:

```
/etc/labelit /dev/xxx/0m fsname volname
```

For example, if you have a cartridge tape drive and you want to make a backup of the /user file system on /dev/dsk/0s7, the **labelit** syntax would be:

```
/etc/labelit /dev/rpt/0m user 0s7
```

See **volcopy** in the *Sys5 UNIX Administrator's Reference Manual*.

Also, before you use **findc**, execute the **find** command. The output of **find** is a list of file names and i-node numbers. **find** is documented in the *Sys5 UNIX User's Reference Manual*. You should be familiar with i-node numbers to use **findc** and **frec**. See "Restoring an Individual File" in this chapter for more information.

---

To recover files written to tape with **finc** or **volcopy**, use **frec**. To use **frec**, the i-node number of the file to be recovered is needed. Use the list of file names and i-node numbers created with **find**. **finc(1M)** and **frec(1M)** are documented in the *Sys5 UNIX Administrator's Reference Manual*.

## Using dd

Use **dd(1)** to back up logical disks containing data bases on raw file systems. **dd** copies the specified input file to the specified output with possible conversions. Note that the standard input and output are used by default. **dd** can also be used in stand-alone at the boot prompt ( : ) as a replacement for **fbackup**. **dd** is explained in the *Sys5 UNIX User's Reference Manual*.

# DISK MIRRORING

---

This chapter explains how to mirror disks. Disk mirroring, which requires two identical physical drives, provides a real-time backup for disk files.

This chapter provides:

- disk mirroring requirements
- examples of disk mirroring
- information describing how to use the **mirutil** utility

See the Appendix in the *Sys5 UNIX Error Message Reference Manual* for a complete listing of **mirutil** error messages.

## Disk Mirroring Overview

The mirrored disk utility, **mirutil**, is available on Plexus systems. To use this utility, you must have two completely identical disk drives (i.e. the same number of heads, sectors/track, etc.). **mirutil** copies information from one drive to a second drive sector by sector, creating mirror images of the same information on two physically separate drives.

Write operations are performed on both disks in the pair. Disk read operations are performed from either disk depending on which drive accesses data first.

---

## Disk Mirroring Requirements and Limitations

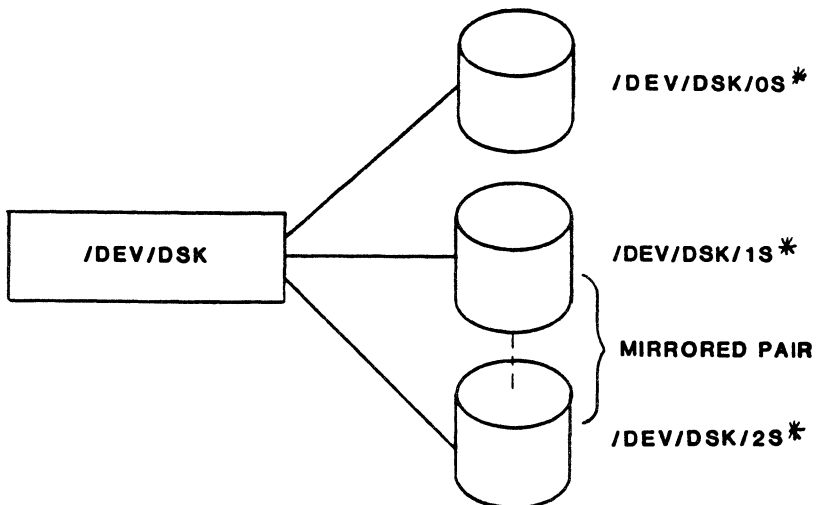
Following are the limitations and requirements for disk mirroring:

- Only entire physical disks (0s0, 1s0, 2s0, ... 15s0) can be mirrored.
- You must be the superuser to invoke **mirutil**.
- More than one pair of drives can be mirrored.
- Only those disks which are accessed via **/dev/dsk** can be mirrored. In a Multibus system where both EMSP (xy) and IMSP (pd) controllers are resident, the **dsk** device defaults to xy. This means if you have EMSP and IMSP controllers in a Plexus system, only EMSP drives can be mirrored.
- Only disks of the same size can be mirrored. Each disk in a mirrored pair must be formatted with the same number of cylinders, sectors, and heads.
- The mirror utility program **/etc/mirutil** runs under the Sys5 kernel not stand-alone.
- The system must be properly shutdown prior to RESET or powering down.
- When the system boots, it reads sector 0 for all **/dev/dsk** devices to determine if any are mirrored. Any disk not running at boot-time will be unavailable. However, a mirrored pair will function as a single drive, if either one is unavailable.
- Logical disks must start at zero offset and include the entire disk. This is set using the stand-alone program, **dconfig**.
- A disk cannot be mirrored with itself.
- The drive to which data is being copied **MUST NOT** contain: rootdev, pipedev, dumpdev, or swapdev.

---

## Example of Disk Mirroring

An example of disk mirroring is illustrated in Figure 12-1. There are three disk drives. Drive 0 is not mirrored and drives 1 and 2 are a mirrored pair. Access to the mirrored pair is only through `/dev/dsk/1s*`, the primary drive.



Not Mirrored: `/dev/dsk/0s0` (Unattached)

Mirrored: `/dev/dsk/1s0` (Primary Drive)  
`/dev/dsk/2s0` (Secondary Drive)

**Figure 12-1.** An Example of Mirrored Disks

When mirroring two drives, any drive can be selected as the primary drive. Mirrored drives do not have to be adjacent to each other (i.e. `/dev/dsk/3s*` can be mirrored to `/dev/dsk/1s*`). Also, the primary drive does not have to be the lower numbered drive.



---

## Using the mirutil Utility

There are two ways to use the **mirutil** utility.

1. You can invoke **mirutil** with the command options (i.e. **mirutil -m 1 2**).
2. You can invoke **mirutil** without command options (i.e. **mirutil**). This action presents a menu from which you can select the desired option.

### Invoking the mirutil utility (with options)

The **mirutil** syntax is:

```
mirutil -option [primarydrive [secondarydrive ]
```

The options are:

- s** status of disks on the system. DO NOT USE the *primarydrive* and *secondarydrive* options with the **-s** option
- m** mirror a pair of disks
- r** re-connect a disconnected disk on an existing pair
- d** disconnect a disk from an existing mirrored pair
- u** un-mirror a pair of disks
- v** verify a mirror pair

*primarydrive* is the drive to be mirrored

*secondarydrive* the drive the primary drive is mirrored on (the invisible disk). This is only needed with the **-m** option.

A few examples of correct syntax follow:

To display the status of disks on the system enter:

```
mirutil -s
```

To mirror **/dev/dsk/1s0** (the primary disk, which is the second drive in the system) on **/dev/dsk/2s0** (the secondary disk, which is the third drive in the system), enter:

---

**mirutil -m 1 2**

To disconnect disk 1 (primary disk) from disk 2 (secondary disk), enter:

**mirutil -d 1** (enter only the primary disk number)

To reconnect disk 1 (primary disk) to disk 2 (secondary disk), enter:

**mirutil -r 1** (enter only the primary disk number)

To verify a mirrored pair enter:

**mirutil -v 1** (enter only the primary disk number)

To un-mirror a pair of disks enter:

**mirutil -u 1** (enter only the primary disk number)

## **Suggestions on How to Use mirutil**

It is recommended to use the **mirutil** menu until you familiar with how **mirutil** operates. After you know the options available, you can perform the same operations available from the **mirutil** menu by invoking **mirutil** with options at the # prompt (i.e. **mirutil -m 1 2**).

---

## Using the Mirror Disk Menu

The following paragraphs describe the mirror disk utility menu options available (see Figure 12-2). To run **mirutil**, do the following:

1. Log in as root (or become the superuser).
2. At the # prompt enter:

**mirutil**

The status of all presently mirrored drives and the mirror utility menu are displayed followed by the mirror disk menu.

| DRIVE | SIZE        | TYPE   | STATUS      | MIRROR | CREATED             |
|-------|-------------|--------|-------------|--------|---------------------|
| 0     | dev/dsk/0s* | 261632 | Independent |        |                     |
| 1     | dev/dsk/1s* | 261632 | Primary     | In use | Drive 2 3:55 7/9 86 |
| 2     | dev/dsk/2s* | 261632 | Secondary   | In use | Drive 1 3:55 7/9 86 |

(S)tatus of disks on the system.

(M)irror a pair of disks.

(R)e-connect a disconnected disk on an existing mirror.

(D)isconnect disks from an existing mirrored pair.

(U)n-mirror a pair of disks.

(V)erify a mirror pair.

(Q)uit.

What do you select? (S/M/R/D/U/V/Q)

**Figure 12-2.** Mirror Disk Menu Display

- 
- (S)tatus Displays the current status of all disks in the system.
- (M)irror Mirrors (assigns) two disks as a mirrored pair.
- (R)e-connect Re-connects a disconnected disk. This is the opposite of Disconnect.
- (D)isconnect Disks Temporarily separates a pair of disks. Disconnect refers to a pair of disks which are set up as a mirror but which are currently not being kept in sync, usually due to some problem such as one of the drives being off-line. The drives are remembered as being mirrored and can be reconnected at any time with Reconnect. In the event of a drive failure of a mirrored disk, a disk becomes disconnected.
- (U)n-mirror Un-mirrors disks from an existing mirrored pair. Un-mirror is the opposite of (M)irror. Unlike disconnect, the un-mirror option forgets all disk mirroring. When disks are un-mirrored each physical disk is a separate drive with no remembered information from disk mirroring.
- (V)erify Checks that a mirrored pair of drives is in agreement. The verify option checks the mirrored pair sector by sector. This will take quite a while. You can use the drive that displays In Use under the Status column when the verify operation is taking place. Use this option if you want to verify that both of your mirrored drives are in agreement.
- (Q)uit Exits **mirutil** and returns the operating system.

- 
3. To select an option from the menu, enter either the upper- or lower-case letter in parentheses (**s**, **m**, **r**, **d**, **u**, **v**, or **q**). As each option is executed, the menu is re-displayed. With the exception of the **(S)tatus** and **(Q)uit** functions, you must have superuser status or login as root.

---

## Mirutil Utility Options

This section describes the **mirutil** options and explains the displays presented by **mirutil**. The options will be discussed in the order they are displayed on the menu.

### Status of Disks on the System

To display the status of all on-line system disks, enter:

**mirutil -s**

OR

Enter:

**mirutil**

The mirror disk menu is displayed (see Figure 12-2). Enter:

**s or S**

The system returns a listing of the status of all system disks by the mirror disk menu (see Figure 12-3).

|   | DRIVE        | SIZE   | TYPE        | STATUS | MIRROR  | CREATED     |
|---|--------------|--------|-------------|--------|---------|-------------|
| 0 | /dev/dsk/0s* | 261632 | Independent |        |         |             |
| 1 | /dev/dsk/1s* | 261632 | Primary     | In use | Drive 2 | 2:12 7/9 86 |
| 2 | /dev/dsk/2s* | 261632 | Secondary   | In use | Drive 1 | 2:12 7/9 86 |

(S)tatus of disks on the system.

(M)irror a pair of disks.

(R)e-connect a disconnected disk on an existing mirror.

(D)isconnect disks from an existing mirrored pair.

(U)n-mirror a pair of disks.

(V)erify a mirror pair.

(Q)uit.

What do you select? (S/M/R/D/U/V/Q)

**Figure 12-3.** Status of Disks on the System

---

Following is an explanation of the **mirutil** display.

**Parameter    Function**

**DRIVE**            Displays each of the 16 possible system drives (/dev/dsk/0s\*  
- /dev/dsk/15s\*).

**SIZE**             Disk size in 512-byte sectors.

**TYPE**             One of three types: Independent (not mirrored), Primary mirror disk, and Secondary mirror disk. The Primary disk is the first named when the pair is connected. It does not have to be the lower numbered disk. For example, **mirutil -m 3 2** mirrors disk 3 on disk 2. Disk 3 is the primary disk. The secondary disk (disk 2) is the invisible disk.

**STATUS**           Blank (nothing in the field), In use, Disconnected, One drive, Unavailable, Mr nnnnn, or Vy nnnnn (where nnnnn is the sector number). A functional independent disk that is not mirrored will show a blank field (nothing in the field). Drives that are mirrored are indicated by In use. Disconnected means that the drive has been disconnected from its mirror. One drive indicates that the other drive in the pair is out of action and this is the survivor. An Unavailable disk is one that does not exist in the system, or is turned off. Mr nnnnn (where nnnnn is the number of sectors) indicates that the disk is in the process of being mirrored. It is currently at the sector number indicated by nnnnn. Vy nnnnn (where nnnnn is the number of sectors) indicates that the disk is in the process of being verified. It is currently at the sector number indicated by nnnnn.

**MIRROR**          Shows which disk is the mirror-pair to the one displayed.

**CREATED**        The time and date the disk was mirrored.

**Figure 12-4.** Explanation of Mirror Disk Status Listing

---

## Mirror a Pair of Disks

Selecting the **m** option (from the Mirror Disk Utility Menu) or invoking the **-m** option (with the **mirutil** command), mirrors (connects) a pair of disks. Before you mirror disks see the section "Disk Mirroring Requirements and Limitations" earlier in this chapter.

To mirror disk 1 (primary) on disk 2 (secondary) enter:

```
mirutil -m 1 2
```

OR

Enter:

```
mirutil
```

The mirror disk menu is displayed (see Figure 12-2). Enter:

```
m or M
```

The following display is returned. What you enter is shown in **boldface** type.



---

| DRIVE | SIZE         | TYPE   | STATUS      | MIRROR | CREATED |
|-------|--------------|--------|-------------|--------|---------|
| 0     | /dev/dsk/0s* | 261632 | Independent |        |         |
| 1     | /dev/dsk/1s* | 261632 | Independent |        |         |
| 2     | /dev/dsk/2s* | 261632 | Independent |        |         |

Enter number of drive with data to be mirrored: 1

| DRIVE | SIZE         | TYPE   | STATUS      | MIRROR | CREATED |
|-------|--------------|--------|-------------|--------|---------|
| 0     | /dev/dsk/0s* | 261632 | Independent |        |         |
| 1     | /dev/dsk/1s* | 261632 | Independent |        |         |

Enter number of drive to overwrite with new data: 2

Ok to copy all data from drive 1 onto drive 2 (y/n)? y

This may take a while and degrade system performance until completed.

- (S)tatus of disks on the system.
- (M)irror a pair of disks.
- (R)e-connect a disconnected disk on an existing mirror.
- (D)isconnect disks from an existing mirrored pair.
- (U)n-mirror a pair of disks.
- (V)erify a mirror pair.
- (Q)uit.

What do you select? (S/M/R/D/U/V/Q)

**Figure 12-5. Connecting a Pair of Mirrored Disks**

---

If you then check the mirror status with the (S) selection from the mirror disk menu or invoke **mirutil -s**, you will see a display similar to Figure 12-6. Note the Mr 6400 in the STATUS column. This shows that the mirror operation is not yet complete. The mirror operation is in the process of copying data from one drive to a second drive.

|   | DRIVE        | SIZE   | TYPE      | STATUS    | MIRROR  | CREATED     |
|---|--------------|--------|-----------|-----------|---------|-------------|
| 1 | /dev/dsk/1s* | 261632 | Primary   | One drive | Drive 2 | 2:12 7/9 86 |
| 2 | /dev/dsk/2s* | 261632 | Secondary | Mr 6400   | Drive 1 | 2:12 7/9 86 |

(S)tatus of disks on the system.

(M)irror a pair of disks.

(R)e-connect a disconnected disk on an existing mirror.

(D)isconnect disks from an existing mirrored pair.

(U)n-mirror a pair of disks.

(V)erify a mirror pair.

(Q)uit.

What do you select? (/S/M/R/D/U/V/Q)

### Figure 12-6. Status After Connecting a Mirrored Pair

It is suggested to select the (S) option from the menu or invoke:

**mirutil -s**

to check the status of disk mirroring.

If:

- you are mirroring two drives
- one drive is the root drive and
- disk mirroring fails

see the section "In Case of Difficulty" in this chapter.

---

## Re-Connect A Disconnected Disk

If either disk in a mirrored pair is non-functional (or for some reason fails during operation) it becomes disconnected. Invoking the **-r** option to the **mirutil** command or selecting **r** from the Mirror Disk Utility Menu re-connects a disconnected disk on an existing mirrored pair.

To re-connect disk 1 (primary drive) to disk 2 (secondary drive) enter:

```
mirutil -r 1 (only the primary drive needs to be entered)
```

OR

Enter:

```
mirutil
```

The mirror disk menu is displayed (see Figure 12-2). Enter:

```
r or R
```

A listing of the primary drives is returned. What you enter is shown in **boldface** type.

---

```

DRIVE      SIZE  TYPE      STATUS      MIRROR  CREATED
1 /dev/dsk/2s* 261632 Primary Disconnected Drive 1 2:33 6/7 86

```

Enter number of drive (primary): 1

This may take a while and degrade system performance until completed.

```

DRIVE      SIZE  TYPE      STATUS      MIRROR  CREATED
0 /dev/dsk/0s* 261632 Independent
1 /dev/dsk/1s* 261632 Primary    Mr 3800    Drive 2 2:33 6/7 86
2 /dev/dsk/2s* 261632 Secondary  One drive Drive 1 2:33 6/7 86

```

(S)tatus of disks on the system.

(M)irror a pair of disks.

(R)e-connect a disconnected disk on an existing mirror.

(D)isconnect disks from an existing mirrored pair.

(U)n-mirror a pair of disks.

(V)erify a mirror pair.

(Q)uit.

What do you select? (S/M/R/D/U/V/Q)

**Figure 12-7. Re-attaching a Detached Disk**

---

**[ NOTE ]** Only primary drives will be listed from which to choose, even if it is the secondary drive that is disconnected.

It is suggested to select the (S) option from the menu or invoke:

**mirutil -s**

to check the status of disk mirroring. When the drives are reconnected, the following display is returned.

| DRIVE          | SIZE   | TYPE        | STATUS | MIRROR  | CREATED     |
|----------------|--------|-------------|--------|---------|-------------|
| 0 /dev/dsk/0s* | 261632 | Independent |        |         |             |
| 1 /dev/dsk/1s* | 261632 | Primary     | In use | Drive 2 | 2:33 6/7 86 |
| 2 /dev/dsk/2s* | 261632 | Secondary   | In use | Drive 1 | 2:33 6/7 86 |

## Disconnecting a Mirrored Pair

At any time you can disconnect (temporarily separate) a disk from a mirrored pair. For example, you might want to disconnect a drive from its mirrored pair if you suspect a drive is not working properly.

If drive 1 (primary) and drive 2 (secondary) are mirrored together and you want to work on drive 1, disconnect drive 1 from drive 2. Both drives contain the same information. While drive 1 is being repaired, use drive 2. Later, after drive 1 is repaired, you can reconnect drive 1 and drive 2 using the Reconnect command. Drive 1 will be updated with the information from drive 2.

---

To disconnect drive 1 (primary drive) from drive 2 (secondary drive) enter:

**mirutil -d 1** (only the primary drive needs to be entered)

OR

Enter:

**mirutil**

The mirror disk menu is returned. Enter:

**d**

A list of mirrored drives is returned. What you enter is shown in **boldface** type. The drive status is displayed after the disconnect operation.

| DRIVE          | SIZE   | TYPE        | STATUS | MIRROR  | CREATED     |
|----------------|--------|-------------|--------|---------|-------------|
| 0 /dev/dsk/0s* | 261632 | Independent |        |         |             |
| 1 /dev/dsk/1s* | 261632 | Primary     | In use | Drive 2 | 2:12 6/7 86 |
| 2 /dev/dsk/2s* | 261632 | Secondary   | In use | Drive 1 | 2:12 6/7 86 |

Enter number of drive to disconnect from its mirror: **1**

Are you sure you want to disconnect drive 1 (y/n) **y**

drive 1 being disconnected from mirror 1.

---

| DRIVE | SIZE         | TYPE   | STATUS      | MIRROR               | CREATED     |
|-------|--------------|--------|-------------|----------------------|-------------|
| 0     | /dev/dsk/0s* | 261632 | Independent |                      |             |
| 1     | /dev/dsk/1s* | 261632 | Primary     | Disconnected Drive 2 | 2:12 6/7 86 |
| 2     | /dev/dsk/2s* | 261632 | Secondary   | One drive Drive 1    | 2:12 6/7 86 |

(S)tatus of disks on the system.

(M)irror a pair of disks.

(R)e-connect a disconnected disk on an existing mirror.

(D)isconnect disks from an existing mirrored pair.

(U)n-mirror a pair of disks.

(V)erify a mirror pair.

(Q)uit.

What do you select? (S/M/R/D/U/V/Q)

**Figure 12-8. Disconnecting a Mirrored Pair**

---

## Un-mirror a Pair of Disks

You can, at any time, un-mirror both disks from a mirrored pair. The un-mirror option forgets all disk mirroring. Select this option only if you want to permanently detach two drives. To un-mirror drive 1 (primary drive) from drive 2 (secondary drive) enter:

**mirutil -u 1**

Note that only the primary drive needed to be entered.

OR

Enter:

**mirutil**

The mirror disk menu is displayed. Enter:

**u or U**

**mirutil** returns the following display. What you enter is shown in boldface type.

| DRIVE          | SIZE   | TYPE    | STATUS | MIRROR  | CREATED     |
|----------------|--------|---------|--------|---------|-------------|
| 1 /dev/dsk/ls* | 261632 | Primary | In use | Drive 2 | 2:12 6/7 86 |

Enter number of drive to unmirror: 1

Are you sure you want to un-mirror drive 1 (y/n) y

After the drives are un-mirrored, **mirutil** returns the following.



---

| DRIVE          | SIZE   | TYPE        | STATUS | MIRROR | CREATED |
|----------------|--------|-------------|--------|--------|---------|
| 1 /dev/dsk/1s* | 261632 | Independent |        |        |         |
| 2 /dev/dsk/2s* | 261632 | Independent |        |        |         |

(S)tatus of disks on the system.  
 (M)irror a pair of disks.  
 (R)e-connect a disconnected disk on an existing mirror.  
 (D)isconnect disks from an existing mirrored pair.  
 (U)n-mirror a pair of disks.  
 (V)erify a mirror pair.  
 (Q)uit.

What do you select? (S/M/R/D/U/V/Q)

### Figure 12-9. Un-mirroring a Mirrored Pair

If you redisplay the mirror status, s, you will see a display similar to Figure 12-10:

| DRIVE          | SIZE   | TYPE        | STATUS | MIRROR | CREATED |
|----------------|--------|-------------|--------|--------|---------|
| 1 /dev/dsk/1s* | 261632 | Independent |        |        |         |
| 2 /dev/dsk/2s* | 261632 | Independent |        |        |         |

(S)tatus of disks on the system.  
 (M)irror a pair of disks.  
 (R)e-connect a disconnected disk on an existing mirror.  
 (D)isconnect disks from an existing mirrored pair.  
 (U)n-mirror a pair of disks.  
 (V)erify a mirror pair.  
 (Q)uit.

What do you select? (S/M/R/D/U/V/Q)

### Figure 12-10. Mirror Status After Un-mirroring a Disk

Only the primary disks are listed. The secondary disk is now accessible.

---

## Verify a Mirror Pair

The verify option is included so the system administrator can verify that two physical drives are in agreement. Use this option to verify that drives are properly mirrored. The verify operation will take a while to complete. You can use the primary disk while the verify operation is being performed.

To do this, invoke the **-v** option (verify) or select **v** from the mirror disk menu. The verify option checks the mirrored pair sector by sector. This option takes a while to complete. For example, the verify operation on a mirrored pair with a 512-byte sector size of 121712 took about one hour. While the verify operation is taking place, you can use the drive with In Use listed under the Status column. While the Verify operation is taking place, drive 1 continues to mirror to drive 2.

To verify that the mirrored pair, drive 1 (primary drive) and drive 2 (secondary drive) are in agreement, enter:

```
mirutil -v 1
```

Note that only the primary drive needed to be entered.

OR

Enter:

```
mirutil
```

The mirror disk menu is displayed (see Figure 12-2). Enter:

```
v or V
```

The program returns a list of primary drives.

---

| DRIVE          | SIZE   | TYPE    | STATUS | MIRROR  | CREATED     |
|----------------|--------|---------|--------|---------|-------------|
| 1 /dev/dsk/ls* | 261632 | Primary | In use | Drive 2 | 2:12 6/7 86 |

Enter number of drive to verify? 1

Are you sure you want to verify drive 1 (Y/N)? Y

| DRIVE          | SIZE   | TYPE        | STATUS | MIRROR  | CREATED     |
|----------------|--------|-------------|--------|---------|-------------|
| 0 /dev/dsk/0s* | 261632 | Independent |        |         |             |
| 1 /dev/dsk/ls* | 261632 | Primary     | In use | Drive 2 | 2:12 6/7 86 |
| 2 /dev/dsk/2s* | 261632 | Secondary   | Vy 270 | Drive 1 | 2:12 6/7 86 |

(S)tatus of disks on the system.

(M)irror a pair of disks.

(R)e-connect a disconnected disk on an existing mirror.

(D)isconnect disks from an existing mirrored pair.

(U)n-mirror a pair of disks.

(V)erify a mirror pair.

(Q)uit.

What do you select? (S/M/R/D/U/V/Q)

**Figure 12-11.** Status While the Verify Operation is Taking Place

---

The verify process will take a while. As the drives are being verified, you can check the status with the (S) option or by entering:

**mirutil -s**

Following is a sample status display.

| DRIVE          | SIZE   | TYPE        | STATUS  | MIRROR  | CREATED     |
|----------------|--------|-------------|---------|---------|-------------|
| 0 /dev/dsk/0s* | 261632 | Independent |         |         |             |
| 1 /dev/dsk/1s* | 261632 | Primary     | In use  | Drive 2 | 2:12 7/9 86 |
| 2 /dev/dsk/2s* | 261632 | Secondary   | Vy 4200 | Drive 1 | 2:12 7/9 86 |

After the verify operation is complete, the status display looks like the following:

| DRIVE          | SIZE   | TYPE        | STATUS | MIRROR  | CREATED     |
|----------------|--------|-------------|--------|---------|-------------|
| 0 /dev/dsk/0s* | 261632 | Independent |        |         |             |
| 1 /dev/dsk/1s* | 261632 | Primary     | In Use | Drive 2 | 2:12 7/9 86 |
| 2 /dev/dsk/2s* | 261632 | Secondary   | In Use | Drive 1 | 2:12 7/9 86 |

## Quit the Mirror Disk Program

To quit the **mirutil** program, select the **q** option from the mirror disk menu. Enter:

**q** or **Q**

The program exits and the operating system is returned.

---

## Shutting Down Your System With Mirrored Disks

To shut down your Plexus system with disks mirrored, invoke the `/etc/shutdown` program. See the chapter *Startup and Shutdown*. You do not need to perform any special shutdown procedures with Sys5 Release 1.5 or later other than the normal `/etc/shutdown`. Startup and Shutdown”.

### In Case of Difficulty

This section describes what to do when disk mirroring fails.

### The System Goes Down While Mirroring Two Drives

If your system goes down while you are performing a mirror operation, such as mirroring two drives, the `mirutil` program recognizes that the drives have been connected. See the chapter *Startup and Shutdown* for information on how to reboot your system. After `/unix` is loaded, the system displays an error message.

In the following example, disk 1 (primary) and disk 2 (secondary) were being mirrored when the RESET button was pressed. After `/unix` was loaded the system returned:

```
A connect to mirror 1 was interrupted,  
run mirutil to try again.
```

---

Invoke **mirutil -s** or invoke **mirutil** and select the (S) option from the menu. You will see a display similar to the following:

| DRIVE | SIZE         | TYPE   | STATUS      | MIRROR     | CREATED             |
|-------|--------------|--------|-------------|------------|---------------------|
| 0     | /dev/dsk/0s* | 261632 | Independent |            |                     |
| 1     | /dev/dsk/1s* | 261632 | Primary     | One drive  | Drive 2 2:12 7/9 86 |
| 2     | /dev/dsk/2s* | 261632 | Secondary   | Disconnect | Drive 1 2:12 7/9 86 |

(S)tatus of disks on the system.

(M)irror a pair of disks.

(R)e-connect a disconnected disk on an existing mirror.

(D)isconnect disks from an existing mirrored pair.

(U)n-mirror a pair of disks.

(V)erify a mirror pair.

(Q)uit.

What do you select? (S/M/R/D/U/V/Q)

### Figure 12-12. System Goes Down While Mirroring Disks

The preceding display shows that a link between drives 1 and 2 has been made but has been interrupted due to the system going down. Invoke **mirutil -r 1** to reconnect drives 1 and 2 or invoke **mirutil** and select the (R) option to reconnect drive 1 to its mirror.

It is suggested to invoke **mirutil -s** or invoke **mirutil** and select the (S) option from the menu to check the status of the reconnect process.

---

## The Root Drive and the Secondary Drive Become Unmirrored

If:

- you mirrored the root drive (primary drive) with a secondary drive (such as drive 1)
- your drives become unmirrored
- and you must repair the primary drive, drive 0

follow the steps below:

1. Shutdown the system. See the chapter *Startup and Shutdown*.
2. Disconnect the drive you need to work on from the system. In this example, disconnect drive 0. Repair the drive.
3. Because drive 0 is disconnected, when you reboot the system, you must tell the boot PROMs to boot from drive 1, the secondary drive. To reboot a P/90 or a P/95 system from drive 1, enter:

**dk(1,0) /unix**

To reboot a Multibus system from drive 1 (EMSP controller)  
enter:

**xy(1,0) /unix**

To reboot a Multibus system from drive 1 (IMSP controller)  
enter:

**pd(1,0) /unix**

For additional information on booting your system, see the chapters *Startup and Shutdown* and *Stand-alone Programs*.

4. After drive 0 has been repaired, shutdown the system and reconnect drive 0 to the system.
5. Boot the system from drive 1 (see step 3).
6. To reconnect drive 0 to drive 1, enter:

---

**mirutil -r 0**

Note that you must specify the primary drive, drive 0, when reconnecting drive 0 and drive 1.





# MANAGING SYSTEM PROCESSES

---

This chapter, which presents an overview of process management, provides explanations of:

- what user processes are
- what **ps(1)** stands for and what happens when **ps** is executed
- how to kill processes
- how to free up frozen terminals
- what thrashing is

---

## Overview

Some of the major functions of an operating system are:

- managing what commands are to be executed.
- managing when commands are executed.
- giving priority to each user's request for processing input to output.

When you enter a command at the C shell ( % ) or Bourne shell ( \$ ) prompt, Sys5 generates a process by assigning a process ID number (PID) to that command and loading it into main memory (RAM).

Most of the process execution that takes place is transparent to the user. As an example, a typical user does not know the priority of the command request or when the job will actually be serviced by the CPU. Sys5 takes care of all that. As the system administrator, you should be aware of processes and how to manage them.

---

## Process Status Command

The command **ps(1)** stands for process status. Use the **ps(1)** command to display listings of system processes. **ps** is fully documented in the *Sys5 UNIX User's Reference Manual*.

The format of the **ps** listing depends on the flags given to the **ps** command. For example, entering:

```
ps
```

returns a display of your processes in short format. Figure 13-1 illustrates a sample display for a user using the C shell and running **troff** and **tbl**:

| PID  | TTY  | TIME | COMMAND |
|------|------|------|---------|
| 1979 | tty0 | 0:12 | csH     |
| 5511 | tty0 | 0:01 | tbl     |
| 5512 | tty0 | 0:00 | sh      |
| 5513 | tty0 | 0:01 | lp      |
| 5514 | tty0 | 0:02 | troff   |
| 5515 | tty0 | 0:00 | post    |
| 5517 | tty0 | 0:01 | ps      |

**Figure 13-1.** A Sample **ps** Listing

A more detailed (and more frequently used version of **ps**) is **ps -ef**, which lists all system processes using a semi-long listing format. A typical **ps -ef** listing will look something like Figure 13-2, which illustrates the results of the **ps -ef** command for an 8-user system in multi-user mode with 6 users logged in. An explanation of the columns of **ps -ef** follows the listing.

---

```

UID PID PPID C STIME TTY TIME COMMAND
root 0 0 2 Jul 2 ? 1266:30 swapper
root 1 0 0 Jul 2 ? 3:58 /etc/init
root 0 0 0 Jul 2 ? 0:02 [ xsched ]
root 3 0 1 Jul 2 ? 10:28 [ lapb ]
root 4 0 0 Jul 2 ? 0:23 [ VT ]
root 5 0 0 Jul 2 ? 0:02 [ DFSL ]
root 6 0 0 Jul 2 ? 0:02 [ DFSR ]
root 7 0 0 Jul 2 ? 0:02 [ mrclean ]
root 6191 1 0 14:51:58 console 0:01 /etc/getty console console
joyce 1979 1 0 18:07:59 tty0 0:14 -csh
root 61 1 0 Jul 2 ? 0:05 /etc/cron
doris 1655 1 0 17:07:29 tty6 0:15 -csh
carol 2225 1 0 03:58:52 vtty0 0:06 -csh
don 1943 1 0 18:03:06 tty2 0:13 -csh
jeffm 1565 1 0 16:48:48 tty3 0:11 -csh
jeffh 1822 1 0 17:28:28 tty4 0:09 -csh
root 6277 1 0 15:05:31 tty5 0:01 /etc/getty /dev/tty5 9600
root 2211 1 0 02:18:58 tty7 0:01 /etc/getty /dev/tty7 1200
lp 72 1 0 Jul 2 ? 0:08 /usr/lib/lpsched
root 5343 1 0 11:16:36 ? 0:01 /etc/getty /dev/vtty1 9600
root 91 1 0 Jul 2 ? 0:01 /etc/getty /dev/vtty2 9600
root 92 1 0 Jul 2 ? 0:01 /etc/getty /dev/vtty3 9600
root 93 1 0 Jul 2 ? 0:01 /etc/getty /dev/vtty4 9600
don 6249 1943 0 14:59:57 tty2 0:00 /bin/sh hold
don 6252 6251 0 14:59:58 tty2 0:19 tbl acp.chap2
don 6253 6251 2 14:59:58 tty2 0:03 [ lp ]
lp 6269 72 0 15:04:53 ? 0:00 /usr/lib/lpsched
doris 6294 1655 0 15:08:34 tty6 0:00 /bin/sh hold3
joyce 6301 1979 5115:09:00 tty0 0:03 ps -ef

```

**Figure 13-2.** A Partial Listing Illustrating the `ps -ef` Command

An explanation of the columns of `ps -ef` follows:

- UID            Username that created the process — the login name.
- PID            Process ID number. A unique number assigned to each job.

---

|         |                                                                                                                 |
|---------|-----------------------------------------------------------------------------------------------------------------|
| PPID    | Parent process ID; the process that spawned this process.                                                       |
| C       | Used by the process for scheduling, priority value.                                                             |
| STIME   | Starting time of process.                                                                                       |
| TTY     | The number of the tty port (terminal) which created the process.                                                |
| TIME    | The cumulative execution CPU time for the process.                                                              |
| COMMAND | The name of the process itself. The full command name and its arguments are printed under the <b>-f</b> option. |

Depending on the flags used with **ps**, other columns might be displayed. For example, executing **ps -elf** displays a longer listing format. See **ps(1)** in the *Sys5 UNIX User's Reference Manual* for additional information.

### What is **getty**?

**getty(1M)**, a process invoked by **init(1M)**, stands for *get tty* program. It runs when no one is logged in and has two arguments. The first argument is the `/dev/tty-number` that it is running on; the second argument is a number indicating the speed (baud rate) at which the port is expecting input.

In the preceding example, the first argument to `/etc/getty` is the tty-number, `/dev/tty5`. The second argument to **getty** refers to the line speed (baud rate), 9600. The speed parameter in `/etc/inittab` must match a label (the first field) in `/etc/gettydefs(4)`. Additional information on the **getty** process is explained in the section on modems in the chapter *Setting Up I/O Devices*. Figure 13-3 illustrates a sample listing of `/etc/gettydefs`.

---

```
console# EXTB HUPCL CS8 OPOST ONLCR # EXTB SANE IXANY TAB3 #Console Login: #console
1200# B1200 HUPCL CS8 # B1200 SANE IXANY TAB3 #Modem login: #300
300# B300 HUPCL CS8 # B300 SANE IXANY TAB3 #Modem login: #1200
9600# B9600 HUPCL CS8 # B9600 SANE IXANY TAB3 #login: #9600
2400# B2400 # B2400 SANE IXANY TAB3 #login: #300
4800# B4800 # B4800 SANE IXANY TAB3 #login: #1200
```

**Figure 13-3.** Sample Listing of `/etc/gettydefs`

---

Each entry in `/etc/gettydefs` has the following format:

`label#initial-flags#final-flags#login-prompt#next-label`

where:

- |               |                                                                                                                                                                                                                                                                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| label         | is the string against which <code>getty</code> tries to match its second argument. It is usually the speed (baud rate) the terminal is supposed to run. You must match this label with the line speed in the <code>/etc/inittab</code> file. See the section on modems in the chapter <i>Setting Up I/O Devices</i> for additional information. |
| initial-flags | are the initial <code>ioctl(1)</code> settings which the terminal is set if a terminal type is not specified to <code>getty</code> . The flags <code>/etc/getty</code> understands are the same as the ones listed in <code>/usr/include/sys/termio.h</code> (see <code>termio(7)</code> ).                                                     |
| final-flags   | These flags take the same values as the initial-flags and are set just prior to <code>getty</code> executing <code>login</code> .                                                                                                                                                                                                               |
| login-prompt  | The entire field, including spaces, is printed as the login prompt.                                                                                                                                                                                                                                                                             |
| next-label    | If this entry does not specify the desired speed (baud rate), indicated by the user typing a <code>&lt;break&gt;</code> character, <code>getty</code> searches the entry with the next-label as its label field and sets up the terminal for those settings.                                                                                    |

See `gettydefs(4)` in the *Sys5 UNIX Programmer's Reference Manual* for more information.

Figure 13-4 illustrates a partial listing of `ps -ef`. Note that the last item, the baud rate parameter to `/etc/getty`, matches a label (first field) in `/etc/gettydefs` (see Figure 13-3). In the `/etc/inittab` file, the baud rate parameter to `getty` matches the label (first field) in `/etc/gettydefs`.



```

root  6191  1  0 14:51:58 console 0:01 /etc/getty console console
root   98   1  0 17:02:01     3   0:00 /etc/getty tty3 9600
root  101   1  0 17:02:01     7   0:00 /etc/getty tty7 1200

```

**Figure 13-4.** Partial Listing of `ps -ef`

As illustrated on the first line above, note that the console baud rate parameter to `/etc/getty` matches the console label in `/etc/gettydefs`. In the second line, the 9600 baud rate parameter to `/etc/getty` matches the 9600 label in `/etc/gettydefs`. In the third line, the 1200 baud rate parameter to `/etc/getty` matches the 1200 label in `/etc/gettydefs`. For additional information see `getty(1M)` in the *Sys5 UNIX Administrator's Reference Manual* and the chapter *Setting Up I/O Devices*.

The `getty` initialization settings are read from the `/etc/gettydefs` text file and implemented when the system is brought to the multi-user state. As such, they are totally configurable (see `initab(5)`, `getty(8)`, and `init(8)` for additional information).

## More About Process Status

When the `who` command is executed, the command returns a listing of ttys currently logged in:

```

joyce  tty0   Jul  7 15:30
jeffm  tty3   Jul  7 09:29
jeffh  tty4   Jul  7 08:33
don    tty2   Jul  7 08:52
doris  tty6   Jul  7 08:13
carol  vtty2  Jul  7 13:23
judy   vtty1  Jul  7 13:53

```

Note that users are logged into ttys 0, 3, 4, 2, and 6 on the local system. Two additional users are logged into virtual terminals (remote system over NOS) `vtty2` and `vtty1`. The following listing illustrates the partial display of the `ps -ef` command. Note that there is a `/etc/getty` for the console, `tty1`, `tty5`, and `tty7` because no one is logged into these terminals.

---

```
UID    PID  PPID  C  STIME   TTY      TIME COMMAND
root   82   1     0  Jul  2   tty1     0:01 /etc/getty /dev/tty1 9600
root  7836  1     0 01:23:06 tty5     0:01 /etc/getty /dev/tty5 9600
root  8203  1     0 08:50:55 tty7     0:01 /etc/getty /dev/tty7 1200
root 11816  1     0 15:41:48 console 0:01 /etc/getty console console
```

As illustrated above, there should be no other processes running for the console, tty1, tty5, and tty7 since users at those terminals and the system console are not logged in to the system to perform any tasks.

Users who are logged in are usually running the Bourne shell (sh) or C shell (csh) for the particular tty number. If -sh or -csh is the only process running for the tty, the user is sitting at the Bourne shell ( \$ ) or C shell ( % ) prompt. The person did not invoke a command when the ps -ef command was run.

---

## Fork and Execute

Each time a user enters a command, the shell makes a copy of itself spawning (or forking) a sub-process with a unique PID number. The parent shell goes into a wait state and the child shell executes the command. When the execution of the child process is completed, the child process exits (or terminates). Control is returned to the parent shell which displays the Bourne shell ( \$ ) or C shell ( % ) prompt again. The name for this task handling sequence is fork and execute. To illustrate this task handling sequence, a user process will be traced in the following paragraph. For additional information, see the chapter *Kernel* in the *Sys5 UNIX System Description* manual.

When a user logs on, after the Bourne shell executes the `.profile` file, the user is given a process dictated by the `/etc/passwd` file (usually a shell). As the user logs in, the `getty` process is replaced with a process for the `login` and subsequently the shell. The user sees the Bourne shell prompt ( \$ ):

Assuming the user's terminal is connected to `tty6`, the process activity looks something like this (in total isolation):

```
UID      PID PPID  C  STIME TTY   TIME COMMAND
doris    50   1   0   Sep 21 6  0:04 -sh
```

Note that this is the only process running for `tty6` — Doris's login shell. If Doris decides to look at one of her files with the `cat` command, she enters:

```
cat filename
```

When the shell receives this command, the shell locates the `cat` program in one of the directories specified in the shell's command `PATH` variable and the parent shell process (PID 50) above duplicates itself into a child shell which executes the `cat` program.

The original parent shell is put into a wait state until the child signals completion. During the time that the `cat` command is outputting the file, `tty6`'s process status would look like:

---

| UID   | PID | PPID | C | STIME  | TTY | TIME | COMMAND |
|-------|-----|------|---|--------|-----|------|---------|
| doris | 50  | 1    | 0 | Sep 21 | 06  | 0:04 | -sh     |
| doris | 52  | 50   | 0 | Sep 21 | 06  | 0:01 | cat     |

Note that the child shell has a unique PID number and also notice the parent process ID PPID. After the `cat` command is finished, the `$` shell prompt returns, the child shell is terminated, and input control to `tty6` is returned to the parent process:

| UID   | PID | PPID | C | STIME  | TTY | TIME | COMMAND |
|-------|-----|------|---|--------|-----|------|---------|
| doris | 50  | 1    | 0 | Sep 21 | 06  | 0:04 | -sh     |

For each command executed by a user, the parent shell forks and executes the child until it either terminates by itself (as in the case with most singular UNIX commands such as `cat(1)`) or the user exits the application, terminates standard input with a `<CTRL-d>`, interrupts or otherwise kills the process. Invoking a database application program, for example, will not terminate the child process until the user signals to the application, in its terms, that (s)he would like to quit. At that time, control is turned back to the parent shell which originally called the program.

The same is true with editors. The parent shell receives the command that invokes the `vi` program file. The user enters the editor. The child is now a process the user is active in while editing the file, and (s)he remains in that process until entering the editor's command to exit (which terminates the child and returns control to the parent).

---

## Killing Processes

Depending on the application or utility, a typical user (without super-user powers) can kill an active child process usually by hitting <CTRL-d> or the interrupt character (usually the <DEL> or <RUBOUT> key on the terminal). This action returns control to the next level parent.

Another way to terminate a process is to display the process table with the **ps** command. This action obtains the PID number. Then, invoke the **kill** command:

```
kill -9 N
```

where:

*N* is the PID number associated with the process

How to invoke the **ps** and **kill** commands will be explained later in the section "Freeing Up A Terminal" in this chapter.

## Who Can Kill Processes?

An ordinary user can kill ONLY those processes associated with his/her UID name. An ordinary user cannot kill others' jobs to protect system security. The **kill** command takes some signal number as its first argument. This tells the system how to kill the process. See **signal(2)** in the *Sys5 UNIX Programmer's Reference Manual* for an explanation of the various signal numbers.

The most often used signal is **-9** which means kill the process no matter what!

The super-user can kill any process. Killing processes is usually performed when a user is stuck using an application program or system command. If, for example, the user is editing a file and all of a sudden the screen freezes and nothing happens on that particular terminal (tty), the user cannot kill the process him/herself, because the terminal is non-functional.

---

## How to Free Up Frozen Terminals

There are many reasons why a terminal might freeze. Following is a list of actions to take. It is suggested to try the actions described in the list in the order presented.

1. Type <CTRL d> or press the <DEL> or <RUBOUT> keys at the user's keyboard.
2. Reset the terminal (see your terminal's manual).
3. The terminal might have received signals (i.e characters that said lock up the keyboard). Power the terminal off. After 30 seconds, turn the terminal on to force a hard reset.
4. Flush the data in the buffers to/from this tty. To do this:
  - a. Disconnect the device from the system port.
  - b. Become the superuser.
  - c. Enter: **stty flushi flusho </dev/ttyxx**  
(xx is the tty number)
5. If only one terminal is frozen and you have tried all the options previously mentioned, you will have to kill the user's processes. See the section "Freeing Up A Terminal" in this chapter.
6. If all the terminals, including the system console, are frozen, see the section "Freeing Up All Terminals" in this chapter.

### Freeing Up A Terminal

In the frequent case where just one tty is stuck, after the the user with the stuck tty notifies the system administrator who has super-user access, the system administrator logs in as root on another terminal, gets a process listing of the system and kills any and all processes associated with the frozen or stuck terminal (tty).

---

```

UID  PID  PPID  C    STIME TTY          TIME COMMAND
root  0     0    2   Jul  2  ?          1266:30 swapper
root  1     0    0   Jul  2  ?           3:58 /etc/init
root  0     0    0   Jul  2  ?           0:02 [ xsched ]
root  3     0    1   Jul  2  ?          10:28 [ lapb ]
root  4     0    0   Jul  2  ?           0:23 [ VT ]
root  5     0    0   Jul  2  ?           0:02 [ DFSL ]
root  6     0    0   Jul  2  ?           0:02 [ DFSL ]
root  7     0    0   Jul  2  ?           0:02 [ mrclean ]
root 6191  1     0 14:51:58 console 0:01 /etc/getty console console
root  61    1     0   Jul  2  ?           0:05 /etc/cron
carol 2225  1     0 03:58:52 vtty0    0:06 -csh
    don 1943  1     0 18:03:06 tty2     0:13 -csh
jeffm 1565  1     0 16:48:48 tty3     0:11 -csh
jeffh 1822  1     0 17:28:28 tty4     0:09 -csh
root  6277  1     0 15:05:31 tty5     0:01 /etc/getty /dev/tty5 9600
root  2211  1     0 02:18:58 tty7     0:01 /etc/getty /dev/tty7 1200
    lp  72    1     0   Jul  2  ?           0:08 /usr/lib/lpsched
root  5343  1     0 11:16:36 ?        0:01 /etc/getty /dev/vtty1 9600
root   91  1     0   Jul  2  ?           0:01 /etc/getty /dev/vtty2 9600
root   92  1     0   Jul  2  ?           0:01 /etc/getty /dev/vtty3 9600
root   93  1     0   Jul  2  ?           0:01 /etc/getty /dev/vtty4 9600
doris 1655  1     0   Jul  2  tty6     0:00 -csh
    don 6249 1943  0 14:59:57 tty2     0:00 /bin/sh hold
    don 6252 6251  0 14:59:58 tty2     0:19 tbl acp.chap2
    don 6253 6251  2 14:59:58 tty2     0:03 [ lp ]
    lp  6269  72    0 15:04:53 ?        0:00 /usr/lib/lpsched
doris 6294 1655  0 15:08:34 tty6     0:00 vi chap2
joyce 6301 1979 5115:09:00 tty0    0:03 vi acp

```

**Figure 13-5.** A Sample Listing Illustrating Process ID Numbers

In Figure 13-5, there are two users currently running the `vi` program: Doris and Joyce. As the system administrator, you must determine whether Doris on `tty6` or Joyce on `tty0` has the stuck process which must be killed. After investigating the problem, you determine that the person using `tty6` (Doris) is the one who is stuck.

---

Logged in as the super-user, you need to execute the **kill** command to terminate all processes associated only with **tty6**. Depending upon your terminal's features, you can also reset the terminal instead of killing the process (the terminal itself might be hung up). See your terminal's manual for additional information.

**[ NOTE ]** Since the PID numbers of child processes are larger than their parents, care must be taken to kill all processes associated with that **tty** in descending numerical PID order so as to gracefully terminate them in sequence:

To kill **tty6**'s processes in descending numerical PID order, enter:

```
kill -9 6294 1655
```

The above action first wipes out the **vi** child process and then proceeds to kill the parent login shell, in effect completely logging Doris off **tty6**. The **login:** message will appear on Doris's terminal. The **ps -ef** command should be run again after the **kill** to verify that both processes were terminated successfully and that only a **getty** is running on **tty6** waiting for login input.

Back at Doris's terminal, if the **kill** was successful, but the screen still appears frozen, the terminal on **tty6** itself probably has to be reset locally or the cabling connections should be checked. Less often it may be necessary to kill a **getty** running on a port that does not display the **login:** message or anything else to the terminal's screen. This is okay to do. The **getty** process, if killed, simply causes **init** to generate a new **getty** process with a different PID number.

Sometimes it is not possible to kill a process associated with a particular terminal. This is usually due to the fact that an I/O operation has not completed.



---

At the end of the day when no users are logged in, there should not be any processes running for any of the numbered ttys, except the getty process itself (meaning all users are off the system). Any other background processes that are associated with a particular tty should be killed. There will be system processes with a ? in the tty field. Leave these processes alone.

Generally, it is good practice to insist that all users be logged off the system before they leave for the day. Plexus systems will not automatically log users off. It is recommended to check to see that no shell processes are running when everyone is gone to protect users' accounts from unauthorized entry.

### **Freeing Up All Terminals and the System Console**

If all the terminals including the system console are frozen, the system might have crashed. In this case (or if killing processes did not help), you must reboot the system and check the file system with **fsck** from the system console. If you have an operable terminal, run **/etc/shutdown** before rebooting the system. See the chapter *Startup and Shutdown* for additional information.

---

## Thrashing

If your system's users generate such demands on system resources (especially on memory and scheduled processes), the system's resources will become taxed. The result might be a degradation of response time. If the demands on the system become extensive, the system can be halted.

There is a limit to the number of processes that the system can handle and this threshold can be reached. Reaching the limits of the system's memory capacity manifests itself as a thrashing. Thrashing is a situation where the operating system scheduler spends much time giving each process its fair turn at the CPU. The time-sharing routines are taking up more of the system's time than other user processes.

If there is a sudden, dramatic slow-down in the system's response time, execute the `ps -ef` command to assist you in verifying processes the system is currently managing. A sudden slow down can occur when someone is doing too much computing. Although rare, this can actually happen when many users are simultaneously executing CPU-intensive utilities (such as `nroff` or `troff`) in the background.

You should know what the output from the `ps` command looks like when the system operates normally. If you execute `ps` and discover that users are simultaneously running many large application/utility processes, tell users that simultaneously running CPU-intensive programs is taxing the system's performance (slowing the system down). You might want to show them how to invoke the `ps -ef` command. Invoking `ps -ef` before they begin a CPU-intensive program, will tell them what other users are doing.

Finally, you might discover that the system has actually been saturated with legitimate processes. Sys5 allows only a fixed number of processes to be executing at one time; this number is installation-dependent. The default is about 100. If you attempt to execute too many processes at once, the system will simply refuse to create any new processes. The shell will indicate this refusal by displaying:

---

cannot fork  
OR  
no more processes

on the system console. Note that the above display will also occur when the swap space needs to be increased. If this occurs, execute the **ps -ef** command to see if someone has created many processes purely by accident. If several attempts to execute **ps** result in these replies or nothing your only alternative is to get another user to: terminate a process or two, exit the editor, log off, etc. This might reduce the number of current active processes and allow you to execute the **ps** and **kill** commands to terminate any and all extraneous processes.

Fortunately, it is only on rare occasions that the system users fill up the process table to this extent by mistake. The system process table can be increased from the default maximum, but care should be taken in doing so. It should be realized that other aspects of system performance can be impacted if the procedure is not performed correctly. To perform this procedure, contact a local guru.

## CRISIS RECOVERY

---

If the power fails without warning when your Plexus system is busy, it can damage the file systems. This section describes how to recover from this damage. First, a Trouble Shooting Guide is presented. The Trouble Shooting Guide tells you what Plexus manual (or chapter of this manual) to read to solve any problems you encounter. Then, the following topics are covered:

- Recovering from Ungraceful Shutdowns
- Recording Memory, ICP and/or ACP Dumps
  - Use this procedure if you have an ICP-based system or a system with ICPs and ACPs.
- Recording Memory and ACP Dumps
  - Use this procedure if you have an ACP-based system.
- Recording Memory and VCP Dumps
  - Use this procedure if you have a VCP-based system, such as a P/95 or a P/90.

---

## Trouble Shooting Guide

Often when a system exhibits failure symptoms, the problem is caused by a simple oversight such as a blown fuse, a switch in the wrong position or a cable not connected correctly. This section provides a checklist to help locate and correct these problems.

For initial system-level troubleshooting, verify the following:

| <u>Check:</u>                                                  | <u>Refer to:</u>          |
|----------------------------------------------------------------|---------------------------|
| <b>System level:</b>                                           |                           |
| System power cord plugged in                                   | <i>Installation Guide</i> |
| System Power On indicator lit                                  | <i>Installation Guide</i> |
| System power keyswitch ON                                      | <i>Installation Guide</i> |
| Main system fuse OK                                            | <i>Installation Guide</i> |
| <b>On tape drive:</b>                                          |                           |
| Tape heads clean                                               | <i>Maintenance Manual</i> |
| Tape cartridge not in SAFE position if trying to write to tape | <i>Maintenance Manual</i> |
| Reel-to-reel write ring installed                              | <i>Maintenance Manual</i> |
| tape controller properly seated                                | <i>Installation Guide</i> |
| <b>On processor module:</b>                                    |                           |
| Baud rate set correctly                                        | <i>Installation Guide</i> |
| Board properly seated                                          |                           |
| <b>On disk drive:</b>                                          |                           |
| Actuator unlocked                                              | <i>Installation Guide</i> |
| Disk-to-controller cables attached                             |                           |

---

## Recovering from Ungraceful Shutdowns

**[ NOTE ]** Attempting to use a damaged file system can cause greater damage to the file system. Use the procedures in this section to protect file systems from further damage.

To shut down the system gracefully when there is advance warning, use the procedures in the chapter, *Startup and Shutdown*.

The act of recovery after an unscheduled system shutdown (such as a power failure) includes three phases. These are

1. Restarting the system
2. Checking and repairing damaged file systems
3. Recovering lost data

These phases of system recovery are discussed in this chapter.

### Restarting the System

After an unexpected power failure, follow this general procedure:

1. Make sure the power switch is in the OFF position.
2. If your system has auto boot enabled, ensure that switch 3 on the switch pack along the front edge of the CPU board is in the OFF position (the fourth switch since the switches are numbered starting with 0). The switch should be in the forward position. To set the autoboot switch OFF, push switch 3 to the forward position with a straightened paper clip.
3. The power should be restored. See the chapter *Startup and Shutdown*. Also, the system should be brought to single-user mode so that the file systems can be checked and repaired. This process is described in the chapter *Startup and Shutdown*.

---

## Repairing File Systems

The program for checking and repairing damaged file systems is **fsck**. It is described in detail in the chapters *Startup and Shutdown* and *Stand-alone Programs*. Also, **fsck** is described in the *Sys5 UNIX Administrator Guide*.

**[ NOTE ]** If the root file system was damaged and you can not reboot, use standalone **fsck** from the boot ( : ) prompt to fix **/dev/dsk/0s1**. If the root file system has more than trivial damage, use **/etc/restor** to restore the file system instead of repairing it.

Before running **fsck**, verify that all file systems to be checked are listed in **/etc/checklist**. This is only necessary if **fsck** is to be used without specifying file systems on a multi-file system installation. The default value for **fsck** with no parameters is the list in **/etc/checklist**.

The command **fsck** should be run either:

- a. once with no parameters, or
- b. once for each file system, with the file system name as the only parameter.

To run the command, enter either:

**fsck** or **fsck /dev/dsk/Xsxx**

where *X* is the physical disk (0 - n)  
*xx* is the logical disk number (0 - 15)

The **fsck** program prompts for information when it finds damage. If you tell **fsck** to remove the file, the file is gone. If you tell **fsck** to restore the file, **fsck** places the file in **lost + found**.

After **fsck** has completed, you might want to recover lost data. To do so, see the next section "Recovering Lost Data".

---

## Recovering Lost Data

When **fsck** is done, depending on the extent of damage, some files and directories might be missing. To recover any erased files, there are two choices:

1. Look in **lost+found**. The **fsck** program puts files here when it can not figure out where they belong. Files in **lost+found** are listed by i-node number such as 000045, 002272, etc.

Copy or move files found there back to their proper locations, then go to step 3.

2. Recover files from backup tapes. In the course of repairing the file systems, **fsck** removes the damaged parts. A good backup or archive system is the best insurance. The procedures for recovering data from **dump** tapes are in the next section of this chapter. The chapter *Backup Procedures* explains how to use **cpio(1)**, **find(1M)**, and **volcopy(1M)**.
3. To return the system to normal multi-user mode, see the chapter *Startup and Shutdown*.

## Running restor

**[ NOTE ]** **dump** and **restor** are Sys3 programs Plexus has included in the Sys5 release. Although instructions describing how to use **dump** and **restor** are included, it is recommended to use **cpio** to make backups and restore files. See the chapter *Backup Procedures* and **cpio(1)** in the *Sys5 UNIX User's Reference Manual* for information on **cpio**.

The purpose for making **dump** tapes is to create copies of file systems on tape, to be used if data on the disk is lost. This can happen when the system is powered off without warning or when users accidentally erase data. Dump tapes are then used to



---

recover the lost data.

Before using **dump** tapes to recover data after an ungraceful shutdown, see the section "Recovering from Ungraceful Shutdowns". In many cases, these procedures can be used to restore the system to its original condition, making the **restor** operation unnecessary.

Restoring part or all of a system from a backup tape is controlled by the command **restor**.

There are two different restore procedures:

- one for restoring single files or directories
- another for restoring entire file systems

Both of these procedures are provided in the subsections that follow.

### **Restoring Files and Directories**

To restore files and directories, do the following:

1. If the data is missing because of a system failure, restore file system integrity as described in the section "Restarting the System".
2. Obtain the latest full **dump** tape and all incremental **dump** tapes taken since the full **dump**. If you know when the file or directory was last changed, use the incremental **dump** tape from that day. Depending on how you make your backups, you might need only the last full **dump** and the last incremental **dump** tape made (see the note below).

---

**[ NOTE ]** It is recommended to make a level 0 **dump** of each file system at least every week. Every evening make a level 9 **dump** of files changed since the last level 0 **dump**. Use a different tape each evening when you make the level 9 **dumps**. If you make incremental dumps using the same **dump** level, you only need two **dump** tapes when you **restor** (the latest level 0 **dump** and the latest incremental **dump**).

3. Insert the latest incremental dump tape into the tape drive.
4. If the tape is not rewound, rewind the tape. Enter:

```
/usr/plx/tape -f /dev/xxx/0m rew  
xxx is rrm for reel-to-reel tape,  
rpt for cartridge tape,  
crm for caching reel-to-reel tape (Multibus), and  
rft for caching cartridge tape (Multibus)
```

Next list the contents of the tape and redirect the output to a file called **temp**. Enter:

```
dumpdir > temp
```

This creates a file **temp** in the current directory. This file contains a list of all the files and directories on the **dump** tape with their i-node numbers.

5. Use one of the following procedures to look for the filename and its i-node number.
  - a. Enter: **more temp** OR
  - b. Use one of the editors (**vi**, **ed**, etc.) to look for the filename and its i-node number in the **temp** listing. OR
  - c. Enter: **grep filename temp**  
(*filename* is the file you are searching for)  
The **grep** program returns the filename and the i-node number.

---

6. If you located the file, go to the next step. If you did not locate the file, unload the tape and remove the tape from the tape drive. Insert the next to the latest incremental tape into the drive and perform steps 4 and 5 to locate the filename. You will need to load incremental **dump** tapes and perform steps 4 and 5 until you find the file you are looking for.

7. After locating the file, change to the directory where you want the file restored.

`cd dirname`

8. Enter:

`/etc/restor x file(s)`

*file(s)* is the name of the files (or directories) to be recovered. Use the entire pathname except for the name of the mounted file system (root directory).

**EXAMPLE:** For the file `/mnt/bin/stuff`, use `bin/stuff`.

9. If the system responds:

`bin/stuff: i-node XXXX`

`Mount desired tape volume: Specify volume #:`

where `XXXX` is the i-node number.

Enter the tape volume number. The volume number is always 1 if the **dump** is only on one tape. If the **dump** is on two or more tapes (because it did not fit on one tape) insert the first volume and enter **2** or the appropriate volume number. If a prompt complains that it can not find the file, load the next volume, etc. When **restor** finds the file, use that volume number.

10. Unload the **dump** tape, remove the **dump** tape from the tape drive, and store it.

11. **restor** restores individual files or directories using i-node numbers for file names. Use the **mv** command to change the i-node number to its file name. For example, if the file in

---

the **temp** listing was **record.c** and the i-node number was 000544, enter:

```
mv 000544 record.c
```

12. Change the file's ownership protections.

If enough of a file system is damaged, it might be easier to restore the entire file system using the procedure described in the section "Restoring Entire Disk File System".

### Restoring Entire Disk File System

Use the following procedure carefully; it overwrites the entire file system. If the root file system is damaged, use the standalone procedures in the chapter, *Stand-alone Programs* to repair it before proceeding. You can seldom repair a root file system after a serious crash because */etc/init* or other files needed to boot the system might be damaged.

To restore an entire file system, do the following:

1. Obtain the latest full **dump** tape and all the incremental **dump** tapes taken since then.
2. You definitely need to restore the latest full **dump** tape (**dump** level 0). Depending on how you make your incremental dumps, the incremental tapes you need to restore will vary. The following table illustrates three methods of making backups. The tapes you need to restore are noted by boldface type.

**Table 14-1.** Different Methods of Making Dump Tapes

|          |   |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|---|
| Method 1 | 0 | 9 | 9 | 9 | 9 | 9 | 9 |
| Method 2 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Method 3 | 0 | 1 | 2 | 1 | 3 | 1 | 2 |

As you can see from the previous table, if you make incremental tapes using Method 1, you need to restore only the latest incremental (level 9) **dump** tape. If you make incremental **dump** tapes using Method 2, you must restore all incremental **dump** tapes. If you make incremental **dumps** using Method 3, you must restore the latest level 1

---

**incremental dump.** This restores all **dump** levels before the level 1 **dump**. Also, you must restore the level 2 incremental **dump** tape.

3. If recovering from a system crash or an ungraceful shutdown, restore file system integrity as described in the section "Recovering from Ungraceful Shutdowns". If recovering accidentally erased data, skip this step.
4. Run the program **mkfs** to create a new, empty file system.
5. Insert the latest full **dump** tape into the tape drive.
6. Enter:

```
restor rf /dev/xxx/0m /dev/dsk/Xsxx
```

where *X* is the name of the physical disk (0 - n)  
*xx* is the name of the logical disk (0 - 15)

EXAMPLE: /dev/dsk/0s3

*xxx* is **rrm** for reel-to-reel tape,  
**rpt** for cartridge tape,  
**crm** for caching reel-to-reel tape (Multibus only), and  
**rft** for caching cartridge tape (Multibus only)

7. Remove the full **dump** tape from the tape drive.
8. Insert the appropriate incremental **dump** tape into the tape drive (see step 2). Enter:

```
restor rf /dev/xxx/0m /dev/dsk/Xsxx
```

where: *X* is the name of the physical disk (0 - n)  
*xx* is the name of the logical disk (0 - 15)

*xxx* is **rrm** for reel-to-reel tape,  
**rpt** for cartridge tape,  
**crm** for caching reel-to-reel tape (Multibus only), and  
**rft** for caching cartridge tape (Multibus only)

9. When the **dump** is finished, remove the tape from the tape drive and store it.

- 
10. Referring to step 2, restore any additional incremental **dump** tapes (see step 8).

Since incremental **dumps** store only files that have changed, all files changed since the last full **dump** are progressively updated to the level of the last incremental **dump**. Files unchanged since the full **dump** remain unchanged.

11. Run **fsck** to check the newly restored file system. See the *Sys5 UNIX Administrator's Reference Manual*.

---

## Recording Memory, ICP, ACP, and VCP Dumps

When a hardware or software problem causes the system to stop in the system debugger, call Plexus Field Service so the hardware problem can be corrected or the software problem diagnosed. If it is not a hardware problem and there is no software solution, Plexus Field Service might request that you take memory and ICP and/or ACP **dumps** to send to Plexus for evaluation.

### Memory, ICP, and ACP Dump Procedure

Follow the steps in this section if you have an ICP-based system or a system with ICPs and ACPs. If you have an ACP-based system, see the section "Memory and ACP Dump Procedure" in this chapter.

**[ NOTE ]** If your system DID NOT crash, but one or more of your ports hung or the system is operating improperly, follow the procedure below to provide more information:

1. Have all users who had normal output from their terminals log off. However, do not have users log off who will change the condition you are reporting.
2. Prepare a snap shot of the process environment by issuing the following command:

```
cd /  
ps -ef > psefl[datetime]
```

where: *[datetime]* are the date and time the command is entered. Do not include any special characters or separators.

3. Update the superblock by entering:

```
sync  
sync
```

4. Continue with the following procedure starting with step 5 — Dump Main Memory.

---

**[ NOTE ]** If your system DID crash or has gone into debugger mode, follow this procedure to prepare the **dump** tape.

1. Write down any error messages exactly as displayed.
2. Press the **<cr>** key and observe whether the console echoes the command.
3. If the error is a bus error and the system reports that it is in SYS DEBUG then enter:

**r 903f00 100**

and write down the last eight bytes (sixteen characters) of the resulting output.

4. If the operating system has reported that it is in SYS DEBUG then type **x** and write down all registers and the PC. Next, type **f** and record all values.
5. Dump the main memory as follows:
  - a. Get a second person to help.
  - b. Press **<RESET>**, enter **sys reset** (68020 job processor systems), or turn the keyswitch to the right, then back to the vertical position. Type an exclamation point (!) immediately after the "PLEX" of the PLEXUS SELFTEST message. You have five seconds to do so.
  - c. If you have a cartridge tape drive, ensure it is set not SAFE. If you have a reel-to-reel tape drive, ensure that the write ring is on the tape. Insert the tape into the tape drive.
  - d. On the console keyboard, enter:

**td(0,0)**

This command dumps the main memory onto the tape. Wait until the memory dump has completed (this will take a while).



---

**[ NOTE ]** Follow the procedure below if your system DID or DID NOT crash.

1. Bring the system up into single-user mode. The autoboot switch must be off. Ensure you have at least 64 blocks per ICP in your root directory to hold the **icpdmpX** files. Dump each ICP with the following command.

```
cd /  
/etc/icpdmp /dev/icX icpdmpX
```

where: *X* is 0, 1, 2, 3, or 4 for ICPs 0, 1, 2, 3, or 4 respectively. This performs an ICP memory dump to the file **icpdmpX**.

See the following example for systems with ICPs and ACPs.

For example if you have a system with 2 ICPs and 2 ACPs, enter:

```
/etc/icpdmp /dev/ic0 icpdmp0  
/etc/icpdmp /dev/ic1 icpdmp1  
/etc/acpdmp /dev/ac2 /<path>/acp2.mem  
/etc/acpdmp /dev/ac3 /<path>/acp3.mem
```

2. Perform this step only if your system crashed. If it did not crash, go to the next step. Position the tape to append to the memory dump without erasing it by entering:

```
/usr/plx/tape srcheof 1
```

3. Position yourself in the root directory by entering:

```
cd /
```

4. Place the **dumps** and other pertinent information onto the tape by entering the first command shown below. If you need to include any ps files you prepared earlier (in step 2 of the first procedure), enter the second command below. **DO NOT** enter both commands.

---

```
ls ./icpdmp* ./<path>/acp?.mem /usr/lib/dnld* /etc/inittab /etc/rc /unix | cpio -oBv >/dev/xxx/0m
```

OR

```
ls ./icpdmp* ./<path>/acp?.mem /psefl* /usr/lib/dnld* /etc/inittab /etc/rc /unix | cpio -oBv >/dev/xxx/0m
```

where:

**xxx** is **rpt** for cartridge tape,

**rrm** for reel-to-reel tape,

**crm** for caching reel-to-reel tape (Multibus only), and

**rft** for caching cartridge tape (Multibus only)

5. Write a short description of the problem you encountered, including:

- What programs were running at the time.
- What devices were attached to the communication processors (e.g., CRTs, printers).
- Any unusual circumstances such as high number of processes, power fluctuations, strange system behavior before the error messages appeared or the crash occurred, room temperatures (such as abnormally warm).

Your description is important for Plexus Field Service to quickly identify the nature of the problem.

---

## Memory and ACP Dump Procedure

This section describes how to perform a memory and ACP dump procedure. Follow the steps in this section if you have an ACP-based system.

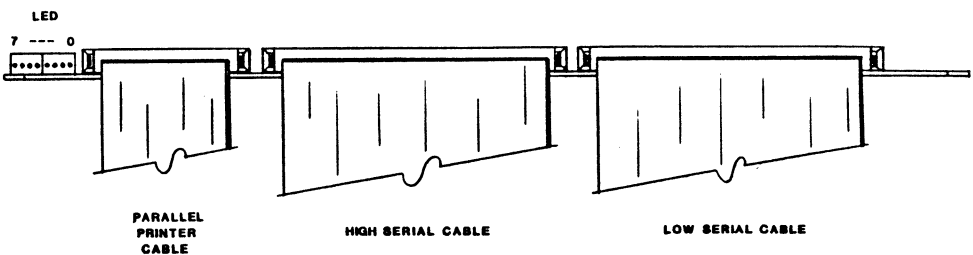
On the ACP there are eight LED indicators on the ribbon connector edge of the board. Four of these indicate:

- ACP waiting for kernel download
- ACP kernel downloading
- ACP kernel running

The two left-most LEDs indicate:

- Bus Error (LED 7, the left-most LED)
- Halt (LED 6, the LED second from the left)

Table 14-2 lists the three ACP states. Figure 14-1 illustrates the LED locations relative to the I/O cables.



**Figure 14-1.** ACP LED Activity Indicators

---

**Table 14-2. ACP LED Status Indications**

---

| LED Position |     |     |     |   |     |     |     | ACP Status                                          |
|--------------|-----|-----|-----|---|-----|-----|-----|-----------------------------------------------------|
| 7            | 6   | 5   | 4   | 3 | 2   | 1   | 0   |                                                     |
| Off          | Off | Off | Off | F | Off | Off | F   | Waiting for kernel download.<br>After system Reset. |
| Off          | Off | Off | F   | F | Off | Off | Off | ACP Kernel up and running OK.                       |
| On           | X   | Off | X   | X | Off | Off | Off | ACP Hard Crash.<br>ACP processor died.              |
| On           | Off | Off | F   | F | Off | Off | Off | ACP Soft Crash.<br>DMA processor died.              |

---

F = Flashing X = indeterminant (either on or off)

**[ NOTE ]** If your system DID NOT crash, but one or more of your ports hung or the system is operating improperly, follow the procedure below to provide more information:

1. Have all users who had normal output from their terminals log off. However, do not have users log off who will change the condition you are reporting.
2. Prepare a snap-shot of the process environment by issuing the following command:

`ps -efl > psefl[datetime]`

where: [*datetime*] are the date and time the command is entered. Do not include any special characters or separators.

3. Update the superblock by entering:

`sync`  
`sync`

---

4. Continue with the procedure below.

**[ NOTE ]** If your system DID crash or has gone into debugger mode, follow this procedure to prepare the dump tape.

The following procedure describes how to perform a memory and ACP dump.

1. Prior to resetting the system after a system/acp crash, record the state of all ACP LED's.
2. Locate the tty connected to port 0 of the ACP and record any system messages printed out (only necessary on ACP's that crashed).
3. If a main memory dump is to be performed, do this now. Follow the procedure below:

- a. Press <RESET>, enter `sys reset` (68020 job processor systems), or turn the keyswitch to the right, then back to the vertical position. Type an exclamation point (!) immediately after the "PLEX" of the PLEXUS SELFTEST message. You have five seconds to do so.
- b. Insert a tape into the tape drive. If it is a cartridge tape, ensure that it is set not SAFE. If it is a reel-to-reel tape, ensure that the write ring is on the tape. Enter:

`td(0,0)`

This command dumps the main memory onto the tape.

- c. Wait until the memory dump has completed (this might take a while).

**[ NOTE ]** Follow the procedure below if your system DID or DID NOT crash.

- 
1. After dumping main memory onto tape, bring the system up into single-user mode. The autoboot switch must be off. Run the utility program `/etc/acpdmp` on all ACPs in the system. Ensure that you have 514 blocks per `acpX.mem` file created.

```
cd /
/etc/acpdmp /dev/ac0 /<path>/acp0.mem
/etc/acpdmp /dev/ac1 /<path>/acp1.mem
/etc/acpdmp /dev/ac2 /<path>/acp2.mem
/etc/acpdmp /dev/ac3 /<path>/acp3.mem
/etc/acpdmp /dev/ac4 /<path>/acp4.mem
```

This action performs an ACP memory dump to the files `acp0.mem`, `acp1.mem`, `acp2.mem`, `acp3.mem`, and `acp4.mem`.

2. Perform this step only if your system crashed. If it did not crash, go to the next step. Position the tape to append to the memory dump without erasing it by entering:

```
/usr/plx/tape srcheof 1
```

3. Position yourself in the root directory. Enter:

```
cd /
```

4. Place the **dumps** and other pertinent information onto the tape by entering the first command shown below. If you need to include any ps files you prepared earlier (in step 2 of the first procedure), enter the second command below. **DO NOT** enter both commands. Specify files using relative path names.

```
ls /<path>/acp?.mem /usr/lib/dnld* /etc/inittab /etc/rc /unix | cpio -oBv > /dev/xxx/0m
```

OR

```
ls /<path>/acp?.mem /psefi* /usr/lib/dnld* /etc/inittab /etc/rc /unix | cpio -oBv > /dev/xxx/0m
```

where:

**xxx** is **rpt** for cartridge drives,

**rrm** for reel-to-reel tape drives,

**crm** for caching reel-to-reel tape (Multibus only), and

**rft** for caching cartridge tape drives (Multibus only)

---

5. Write a short description of the problem you encountered, including:

- What programs were running at the time.
- What devices were attached to the communication processor (e.g., CRTs, printers).
- Any unusual circumstances such as high number of processes, power fluctuations, strange system behavior before the error messages appeared or the crash occurred.

Your description is important for Plexus Field Service to quickly identify the nature of the problem.

### Contents of Memory and ACP Dump Tapes

If you intend to send dump tapes to Plexus for evaluation, the tapes must contain the following:

- All dump tapes should have the system memory dump as file 1.
- File 2 of dump tapes should be a **cpio** of the following files.

```
./unix
./usr/lib/dnld/acp
./usr/lib/dnld/acp5.syms
./<path>/acp0.mem
./<path>/acp1.mem
./<path>/acp2.mem
./<path>/acp3.mem
./<path>/acp4.mem
./etc/inittab
./etc/rc
```

- Provide written information pertaining to ACP LED's at time of system/acp crash:
  1. ACP Port 0 messages, if any
  2. System console messages, if any
  3. Register contents, if possible

---

## Memory and VCP Dump Procedure

This section describes how to perform a memory and VCP dump procedure. Follow the steps in this section if you have a P/90, P/95 or other VCP-based system.

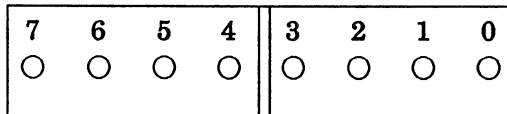
On the VCP there are eight LED indicators on the ribbon connector edge of the board, numbered 7 through 0. Four different states of the eight LEDs indicate:

- VCP waiting for kernel download
- VCP kernel up and running OK
- DMA bus error
- 68020 bus error

The two left-most LEDs indicate:

- (LED 7, the left-most LED) — Bus Error
- (LED 6, the LED second from the left) — Busy (when the LED blinks on and off very fast) or Error (when lit)

LEDs 5 through 0 are software programmable. Figure 14-3 illustrates the VCP LED activity locations. Table 14-3 lists the three VCP states.



**Figure 14-3.** VCP LED Activity Indicators



---

**Table 14-3. VCP LED Status Indicators**

| LED Position |     |     |     |   |     |     |     | VCP Status                   |
|--------------|-----|-----|-----|---|-----|-----|-----|------------------------------|
| 7            | 6   | 5   | 4   | 3 | 2   | 1   | 0   |                              |
| Off          | Off | Off | Off | F | Off | Off | F   | Waiting for kernel download  |
| Off          | Off | Off | F   | F | Off | Off | Off | VCP kernel up and running OK |
| On           | Off | Off | F   | F | Off | Off | Off | DMA bus error                |
| On           | X   | Off | X   | X | Off | Off | Off | 68020 bus error              |

F = Flashing X = indeterminant (either on or off)

**[ NOTE ]** If your system DID NOT crash, but one or more of your ports hung or the system is operating improperly, follow the procedure below to provide more information:

1. Have all users who had normal output from their terminals log off. However, do not have users log off who will change the condition you are reporting.
2. Prepare a snap-shot of the process environment by issuing the following command:

**ps -efl > psefl[datetime]**

where: [datetime] are the date and time the command is entered. Do not include any special characters or separators.

3. Update the superblock by entering:

**sync**  
**sync**

4. Continue with the procedure below.

---

[ **NOTE** ] If your system DID crash or has gone into debugger mode, follow this procedure to prepare the **dump** tape.

The following procedure describes how to perform a memory and VCP dump.

1. Prior to resetting the system after a system/vcp crash, record the state of all VCP LED's.
2. Write down any console error messages exactly as displayed.
3. Locate the tty connected to port 0 of the VCP and record any system messages printed out (only necessary on VCP's that crashed).
4. If the operating system has reported that it is in SYS DEBUG, then type **x** and write down all registers and the PC. Next, type **f** and record all values.
5. If a main memory dump is to be performed, do this now. Follow the procedure below:
  - a. Press <RESET>, enter **sys reset** (68020 job processor systems), or turn the keyswitch to the right, then back to the vertical position. Type an exclamation point (!) immediately after the "PLEX" of the PLEXUS SELFTEST message. You have five seconds to do so.
  - b. Insert a tape into the tape drive. If it is a cartridge tape, ensure that it is set not SAFE. If it is a reel-to-reel tape, ensure that the write ring is on the tape. Enter:

**td(0,0)**

This command dumps the main memory onto the tape.

- c. Wait until the memory dump has completed (this might take a while).

---

**[ NOTE ]** Follow the procedure below if your system DID or DID NOT crash.

1. After dumping main memory onto tape, bring the system up into single-user mode. The autoboot switch must be off. Run the utility program `/etc/vcpdmp` on all VCPs in the system. Ensure that you have 1200 blocks per `vcpX.mem` file created.

```
cd /
/etc/vcpdmp /dev/vc0 /<path>/vcp0.mem
/etc/vcpdmp /dev/vc1 /<path>/vcp1.mem
/etc/vcpdmp /dev/vc2 /<path>/vcp2.mem
/etc/vcpdmp /dev/vc3 /<path>/vcp3.mem
```

This action performs an VCP memory dump to the files `vcp0.mem`, `vcp1.mem`, `vcp2.mem`, and `vcp3.mem`.

2. Perform this step only if your system crashed. If it did not crash, go to the next step. Position the tape to append to the memory dump without erasing it by entering:

```
/usr/plx/tape srcheof 1
```

3. Position yourself in the root directory by entering:

```
cd /
```

4. Place the dumps and other pertinent information onto the tape by entering the first command shown below. If you need to include any ps files you prepared earlier (in step 2 of the first procedure), enter the second command below. DO NOT enter both commands. Specify files using relative path names.

```
ls /<path>/vcp?.mem /usr/lib/dnld* /etc/inittab /etc/rc /unix | cpio -oBv > /dev/xxx/0m
```

OR

```
ls /<path>/vcp?.mem /pseff* /usr/lib/dnld* /etc/inittab /etc/rc /unix | cpio -oBv > /dev/xxx/0m
```

---

where:

**xxx** is **rpt** for cartridge drives,  
**rrm** for reel-to-reel tape drives,  
**crm** for caching reel-to-reel tape (Multibus only), and  
**rft** for caching cartridge tape drives (Multibus only)

5. Write a short description of the problem you encountered, including:
- What programs were running at the time.
  - What devices were attached to the communication processor (e.g., CRTs, printers).
  - Any unusual circumstances such as high number of processes, power fluctuations, strange system behavior before the error messages appeared or the crash occurred.

Your description is important for Plexus Field Service to quickly identify the nature of the problem.

### Contents of Memory and VCP Dump Tapes

If you intend to send dump tapes to Plexus for evaluation, the tapes must contain the following:

- All dump tapes should have the system memory dump as file 1.
- File 2 of dump tapes should be a **cpio** of the following files.

```
./unix
./usr/lib/dnld/vcp
./usr/lib/dnld/vcp4.syms
./<path>/vcp0.mem
./<path>/vcp1.mem
./<path>/vcp2.mem
./<path>/vcp3.mem
./etc/inittab
./etc/rc
```

- 
- Provide written information pertaining to VCP LED's at time of system/vcp crash:

1. VCP Port 0 messages, if any
2. System console messages, if any
3. Register contents, if possible

# STAND-ALONE PROGRAMS

---

This chapter describes what stand-alone programs do and explains how to use them. This chapter discusses the stand-alone programs that are not documented elsewhere in Plexus manuals.

## Overview

Stand-alone programs run independently of Sys5. They are designed to run directly from the boot level, signified by the boot prompt ( : ). After pressing RESET, entering `sys reset` (68020 systems), or turning the keyswitch to the right, then back to the vertical position, the system performs a self-test. Next, the boot prompt ( : ) is displayed. To invoke a stand-alone program at the boot prompt, you must specify the device and location of the stand-alone program you want to run. The details and syntax of this are explained later in this chapter.

The capability to run programs independently from Sys5 is sometimes required to create, alter, or repair the environment in which Sys5 runs. Under certain circumstances you could not change the Sys5 environment without disabling the operating system kernel (and your accumulated files and programs) at the same time. Programs of this category alter or maintain the organization and/or format of the disk drive(s).

Some Sys5 stand-alone utilities are provided, not to alter the environment, but to check system status of various parameters while the system is down. Others provide device-to-device copying, backup, and/or restore capabilities when it is inconvenient or impossible to bring the system up to single-user or multi-user mode first.

---

Although you will frequently run the Sys5 versions of several stand-alone programs described in this chapter such as **cat** and **ls**, you will rarely find it necessary to invoke the stand-alone programs.

You will definitely need to invoke stand-alone programs if you:

- Install a new disk
- Experience a catastrophic disk failure
- Increase the swap area or change its location
- Spare some bad disk sectors (for EMSP, AMSP, or P/90 disks) or tracks (for IMSP disks)
- Create additional file systems (after **/dev/dsk/0s2** - the swap area)
- Change the default logical disk sector addresses on the disk

Notice that most of these examples are once-per-system occasions.

---

## Plexus Release Tape Stand-alone Programs

Plexus release tapes contain some or all of the following files. The number in parentheses before the name indicates the position of the file (file number) on the release tape.

- (0) **help** Gives information about the release tape and how to use stand-alone programs.
- (1) **RESERVED**
- (2) **unix** A binary copy of the Sys5 operating system kernel.
- (3) **dformat** Disk format. Formats a disk drive, prompts for format information, and enables sparing of bad disk sectors.
- (4) **mkfs** Make file system. Creates and organizes a bare file system. See also **mkfs(1M)** in the *Sys5 UNIX Administrator Reference Manual* and in the chapter *Configuring Disks*.
- (5) **restor** Restores a file system from **dump** tape. Use stand-alone **restor** for **/dev/dsk/0s1**. See also **/etc/restor** in the chapter *Backup Procedures*.
- (6) **fsck** Interactive file system consistency check. See also **fsck(1M)** in the *Sys5 UNIX Administrator's Reference Manual*.
- (7) **dd** Device-to-device copy program. Can read out of file systems. See **dd(1)** in the *Sys5 UNIX User's Reference Manual*.
- (8) **fbackup** Fast backup. Does a quick image copy from disk to tape or from tape to disk on Multibus systems. On VMEbus-based systems, file number 8 is RESERVED.
- (9) **od** Octal dump. Dumps a file to the screen in octal or hex format. See **od(1)** in the *Sys5 UNIX User's Reference Manual*.
- (10) **dconfig** Configures a disk with initialization information and the logical file system layout for Sys5. See also the chapter *Configuring Disks* in this manual.



---

(11) RESERVED

(12) **fsdb** \* File system debugger. See **fsdb**(1M) in the *Sys5 UNIX Administrator's Reference Manual*.

(13) **du** \* Reports disk usage. See **du**(1) in the *Sys5 UNIX User's Reference Manual*.

(14) **ls** \* Lists contents of directories. See **ls**(1) in the *Sys5 UNIX User's Reference Manual*.

(15) **cat** \* Concatenates and prints files. See **cat**(1) in the *Sys5 UNIX User's Reference Manual*.

A few general remarks about stand-alone programs:

1. Never run the stand-alone version of a program while Sys5 is booted. You must always run it from the boot level ( : ) when the operating system kernel, **/unix**, is not loaded.
2. Stand-alone programs cannot create or write to regular disk files, although some can write special files (devices such as tape or disk).
3. Stand-alone programs do not allow redirection, pipes, or pattern matching on the command line.
4. If you make a mistake when entering the program name, press # or <backspace> to erase one character. To erase one line, press @ or <CTRL x>.
5. The initial access to cartridge tape rewinds it before reading or writing it.
6. Writing to a cartridge tape must be in multiples of 512 bytes.
7. Stand-alone programs automatically buffer read requests from special files; the buffer size is 1024 bytes. Thus, read

---

\* These stand-alone programs are in the **/stand** directory on disk, but might not be included in your tape copy of stand-alone programs.

---

requests for less than 1024 bytes result in 1024 bytes being allocated per read.

8. Writing to any of the special disk files must be in blocks of 1024 bytes per write.
9. The stand-alone commands normally use `/dev/dsk/0s1` as the root file system. This can be changed by changing the `rootdev`, `pipeudev`, `dumpdev`, and `nswap` values in block 0 of `/dev/dsk/0s1`. See the chapter *Changing Root and Swap* in this manual.

---

## How to Run a Stand-alone Program

**[ NOTE ]** Stand-alone programs are self-contained and do not run under any operating system. They can be loaded and executed when Sys5 has not been loaded into memory. To abort a running stand-alone program, press DELETE or RUBOUT.

The computer hardware and firmware (boot PROMs) control access to these programs. Your system is at the boot level and ready to receive instructions when the console displays

```
PLEXUS PRIMARY BOOT REV X.X COMPLETE  
:□
```

where □ represents the cursor.

Because the amount of memory at the PROM level is limited, commands issued at the PROM level are very rudimentary. For example, PROMs cannot find and execute a program unless you specify exactly where that program is. This can be done one of two ways:

1. The longer method. With the longer method, you specify the device and location using boot level naming conventions for the devices.
2. The shorter method uses a shorthand syntax built into the PROMs that distinguishes only between the system's default disk drive and its default tape device.

The longer command format is useful if you have more than one backup device and need to specify which one has the stand-alone programs you want to use. For example, if you have two tape drives or if you have two hard disks and keep a second copy of the stand-alone directory on the second one, use the longer command format. It also allows you to keep and access more than one program of the same name on a tape.

---

## The Long Method

This section describes the long method for loading stand-alone programs from disk and tape.

### Loading Stand-alone Programs from Disk

To invoke stand-alone programs that reside on disk at the boot prompt ( : ) using the long method enter:

*device (number,offset) program\_name*

where:

*device* is **pd** (IMSP disks), **xy** (EMSP disks), **dk** (P/90 and P/95).

*number* is the physical disk number (0 is the first physical disk, 1 is the second physical disk, etc.)

*offset* is the starting sector address of the file system that contains the program.

*program\_name* is the complete pathname (without the initial slash in this file system).

To call the stand-alone program **mkfs** from the **/stand** directory in the root file system on your primary disk (IMSP drive), at the boot prompt ( : ), enter:

**pd(0,0)stand/mkfs**

If you moved your stand-alone directory to the **/dev/dsk/0s3** file system, which begins at 100000 (512-byte sectors) you instead enter:

**pd(0,100000)stand/mkfs**

If you moved (or copied) your stand-alone directory onto the first file system on a second hard disk (i.e., disk 1), and named that directory **bak.stand**, enter:

**pd(1,0)bak.stand/mkfs**

---

In all cases, the system responds:

\$\$mkfs □

where □ represents the cursor.

### Loading Stand-alone Programs from Tape

**[ NOTE ]** Boot ROM cannot perform byte swapping. The stand-alone boot tape received from Plexus is byte swapped.

To invoke stand-alone programs that reside on tape at the boot prompt ( : ) using the long method enter:

*device (number,offset)*

where:

*device* is the tape device name (**rm** for reel-to-reel tape drives, **pt** for cartridge drives).

*number* is the physical tape device number (0 for the first tape drive, 1 for the second tape drive, etc.)

*offset* is the number of the file that contains the program you want to run. The file numbers are listed toward the beginning of this chapter.

For example, to run stand-alone **fsck** (file number 6) from your release tape cartridge, respond to the boot level prompt ( : ) by entering:

**pt(0,6)**

---

The system responds:

\$\$fsck

where  represents the cursor.

The following example describes how to run stand-alone **fsck** from your reel-to-reel tape drive. To run stand-alone **fsck** (file number 6) from your release tape, respond to the boot level prompt ( : ) by entering:

**rm(0,6)**

The system responds:

\$\$fsck

where  represents the cursor.

For tape drives, you do not specify the *program\_name* because there is only one program at that location.

## The Short Method

Plexus systems permit truncating both the tape and the disk-program names; the slash ( / ) as the first character in the path-name means the program is disk-resident.

Thus, to execute the stand-alone program, **mkfs**, from disk, at the boot prompt ( : ) enter:

**/stand/mkfs**

The slash ( / ) indicates that **stand** is on the disk.

To execute the stand-alone program, **mkfs**, from the release tape, only the program name must be entered in response to the boot prompt ( : ). To execute the tape resident program, enter:

**mkfs**

Most of the stand-alone programs have counterparts described in the *Sys5 UNIX Administrator Reference Manual* and the *Sys5 UNIX User's Reference Manual*. When called, the stand-alone programs all return a double dollar sign (\$\$) followed by the program name. The cursor is positioned one space to the right of the

---

program name. You enter arguments to the program there. For example, if you enter:

```
pd(0,0)stand/dd or /stand/dd (IMSP drives)
xy(0,0)stand/dd or /stand/dd (EMSP drives)
dk(0,0)stand/dd or /stand/dd (P/95, P/90 drives)
```

the system responds

```
$$ dd 
```

where  represents the cursor.

At the  prompt, enter the arguments to the **dd** program. Arguments to stand-alone programs are usually the same as non-stand-alone versions and are described in the *Plexus Sys5 UNIX User's Reference Manual*. Some stand-alone programs are interactive; that is, they prompt for any parameters needed.

Upon completion, stand-alone programs print Exit followed by a value. The value 0 (zero) indicates that the program completed normally.

## Arguments to Stand-alone Programs

The following special files can be used as arguments to the stand-alone commands.

**[ NOTE ]** Note that these device names can be used only after a stand-alone program has been loaded and are not to be confused with boot-level device names.

**/dev/dsk/XsXX** Logical disks on the designated physical disk(s). *X* is the physical disk number. *XX* is the logical disk number, ranging from 0 to 15. On a system with both IMSP and EMSP controlled disks, **/dev/dsk/XsXX** defaults to EMSP controlled disks. To access an IMSP controlled disk, use **/dev/pd/XsXX**. For more information on logical disks, see the the chapter

---

*Configuring Disks.*

- /dev/pd/XsXX** Used to access logical disks on IMSP controlled physical disks (systems with IMSP and EMSP controllers). If you have a system with IMSP and EMSP controlled disks, the system defaults to EMSP-controlled disks. To access an IMSP-controlled disk, use **/dev/pd/XsXX**. *X* is the physical disk number.
- /dev/xyXsXX** Used to access logical disks on EMSP-controlled physical disks.
- /dev/rpt/0m[n]** Cartridge tape drive (raw device). 0m is rewind and 0mn is no rewind.
- /dev/rpt/0fm[n]** Cartridge tape drive (bit flipped). This is a raw device. 0fm is rewind and 0fmn is no rewind (P/95 only).
- /dev/rrm/0m[n]** Reel-to-reel tape drive (raw device). 0m is rewind and 0mn is no rewind.
- /dev/rrm/0bm[n]** Reel-to-reel tape drive with byte swapping (raw device). 0bm is rewind and 0bmn is no rewind.
- /dev/null** The bit bucket.
- /dev/console** The console.



---

## The Mount and Skip Commands

The stand-alone commands recognize two special built-in commands that set up their environment: **mount** and **skip**. **mount** allows access to disk files on file systems other than **/dev/dsk/0s1**. To use it, type an exclamation point (!) followed by a **mount** command in response to the stand-alone prompt **\$\$**. For example, to display files on a file system on **/dev/dsk/0s2** with the **cat** command, (user response in **bold**):

```
$$ cat ! mount /dev/dsk/0s2 /fs2
<CTRL d>
$$ cat /fs2/user/foobar
```

Note that entering **<CTRL d>** causes a reprompt of **\$\$** followed by the command name.

The **skip** command allows direct access to particular files on tape; it positions the tape at the beginning of the file number you specify. To use **skip**, type an exclamation point (!) followed by the **skip** command in response to the stand-alone prompt **\$\$**. For example, to display cartridge tape file 3 with the **cat** command (user response in **bold**):

```
$$ cat ! skip /dev/rpt/0mn 3
<CTRL d>
$$ cat /dev/rpt/0mn
```

Note that entering **<CTRL d>** causes the reprompt **\$\$** followed by the name of the command. Note also that the **skip** command requires **/dev/rpt/0mn** (rather than **/dev/rpt/0m**). Using **/dev/rpt/0m** rewinds the tape before the second prompt.

---

## Descriptions of Stand-alone Programs

This section describes the functions of the stand-alone programs. These programs will be discussed in the order they appear on the release tape. Stand-alone programs are available on disk in the **/stand** directory and on your release tape. Invoking stand-alone programs from disk is faster and easier than loading stand-alone programs from your release tape.

### help

Information about the release tape is displayed on the system console by the **help** program. It is self-explanatory and uses menus to allow access to more detailed information. To use the help program:

1. Insert the release tape into the tape drive
2. At the boot prompt (:), enter:

**help**

To use the help program on disk, enter:

**/stand/help**

### unix

Under normal circumstances, this copy of the Sys5 operating system is never accessed. It is included only as a last-resort backup in case all other copies have been lost. To load the kernel, **unix** from tape do the following:

1. Insert the release tape into the tape drive.
2. At the boot prompt (: ) enter:

**unix**

The kernel, **unix**, is loaded, then the # prompt appears. See the chapter *Startup and Shutdown* for additional information.

---

## **dformat**

The controlling program for formatting hard disks is **dformat**. Formatting involves encoding the disks' magnetic surface with boundaries so the operating system can find byte patterns to read or blank spaces to write to. Each disk is divided (encoded) into tracks and sectors. Tracks are magnetic patterns of concentric circles on the disk surfaces. Tracks are divided into sectors.

To the user, sectors have 512 bytes each. But actually, **dformat** knows that each sector is larger than 512 bytes by about 15%. This larger figure is known as the physical bytes per sector, and it varies according to the disk controller and disk drive installed in your system. For example, a disk might actually have 586 bytes per sector. The extra 72 bytes (beyond 512) provide the disk controller hardware and firmware the pointers and addresses it needs to mark the beginning sector data.

A microscopic flaw in the magnetic coating on a disk can prevent the operating system from reading or writing to the track(s) that share disk space with that flaw. When this happens, that track must be spared. That is, to keep the disk usable, the operating system must be informed that that track is bad. Then it can set up a new area of the disk to substitute for the bad part, remap the disk to skip the bad and use the substitute track.

The stand-alone program, **dformat**, performs both types of disk preparation and maintenance routines: initial setup and formatting, and sparing, either automatic or manual.

The main purpose of **dformat** is to set sector size, number of sectors per track, number of alternate tracks or sectors, and to spare out the known bad sectors. Like most operating systems, Sys5 can only access its disk by sectors. It must know exactly how large these sectors are in order to transform block requests into sector requests. Once the sector size and other issues have been decided, **dformat** goes out and defines sectors on the disk, writing sector headers for each sector. These headers make the sector able to identify itself to the disk controller and ultimately to Sys5 when the sector is accessed, so the system can be sure it is getting the right one.

---

**dformat** can also perform disk tests and repairs. It finds out if the disk can write data, and also if the disk can read what it writes. If **dformat** finds a track it can not read or write, it automatically spares it. It sets up a bad-track table such that any requests to access a bad track are automatically redirected to a good track. You can also use **dformat** to manually spare any bad tracks reported by a Sys5 error message.

**dformat** can accept disk information either manually or automatically. You can make a tape with your disk information on it and use this tape to run **dformat**, or you can enter the information interactively.

**dformat** has several options. Not every option is available on every Plexus machine.

### Executing dformat

When you run **dformat**, you will first see a menu of options. Select the letter of the option (f, s, r, l, v, or q) you wish to run. These are:

#### **f** Format the disk.

The formatting operation has two phases: format and test. The format part consists of writing sector headers and writing the user pattern on the disk. The test portion consists of reading the user pattern just written, looking for unreadable sectors, and sparing any unreadable tracks it finds. With IMSP disk controllers, error messages of the form:

```
track <cylinder>/<head> spared
```

are printed for all bad tracks found and spared.

#### **F** Format (No delay in listing bad sectors).

This option is used at the factory to format disks.

#### **s** Spare tracks.

Enables manual (i.e., operator intervened) sparing of tracks or sectors. Be sure to make a new **dformat** bad track tape using the **l** option after you spare any newtracks or sectors.

---

**r Read the disk, performing only a surface analysis.**

Does not re-format or spare bad tracks. This option generates and prints a report of the bad tracks on the disk, in the form:

```
bad trk <cylinder>/<head>
```

**l List bad tracks.**

This option prints a table of all the bad tracks or sectors and their corresponding spares, in decimal and hex. You can also use this option to make a bad track tape for your disk, to use if you must reformat the disk. It is a very wise idea to make this bad track tape right after you format your disk and keep it in a safe place. Be sure to remake the tape if you spare any additional sectors!

**v Verify a single sector (EMSP, AMSP, and P/90 controllers)**

This option allows you to test a single sector. It writes to and reads from the specified sector, notifying you of any errors. This option makes the data on this sector invalid; therefore, **DO NOT** use this option unless you are pretty sure the sector is defective. The file using the tested sector will no longer be valid!

**q Quit the program.**

**dformat** spares disks controlled by the IMSP by whole tracks. If there is a bad area on the disk, it remaps to one of twenty alternate cylinders. Disks controlled by an EMSP are spared by sector. The controller reserves two sectors at the end of each track as alternates. Disks controlled by a SCSI are spared by sector. For disks controlled by an AMSP, if only one sector on a track is bad, that sector is spared. If more than one sector is bad, the entire track is remapped to one of 50 alternate tracks.

**dformat** with the **f** option rewrites the disk's sector headers. These are written by Plexus in the factory and should never have to be rewritten. The **f** option should be used only after consultation with Plexus field service (hardware maintenance) personnel.

---

In most cases, no matter how badly the file system has been scrambled, the disk can be completely restored by the use of **dconfig**, **mkfs**, and a backup restoration utility such as **restor** in that order.

Because **dformat** with the **f** option writes a pattern of bits all over the disk, any previous contents of the disk are lost. If the disk still has good data on it, back it up before running **dformat** with the **f** option.

The default sector size is 512 bytes; the default block size is 1024 bytes.

The following table lists the total number of 512-byte sectors normally created by **dformat** on Plexus disks. This table also lists the total number of 1K-byte blocks, which is half the number of 512-byte sectors.

---

**Table 15-1. Total Number of Sectors Created by dformat**

| <b>Disk<br/>Size</b> | <b>Controller<br/>Name</b> | <b>Total 512-byte<br/>Sectors</b> | <b>Total 1K-byte<br/>Blocks</b> |
|----------------------|----------------------------|-----------------------------------|---------------------------------|
| NEC 22Mb 8"          | IMSP                       | 40046                             | 20043                           |
| NEC 36Mb 8"          | IMSP                       | 66810                             | 33405                           |
| Fuj. 71Mb 8"         | IMSP                       | 134946                            | 67473                           |
| Fuj. 72Mb 14"        | IMSP                       | 136170                            | 68085                           |
| NEC 142Mb 8"         | IMSP                       | 272544                            | 136272                          |
| Fuj. 145Mb 8"        | IMSP                       | 273020                            | 136510                          |
| Fuj. 145Mb 14"       | IMSP                       | 272340                            | 136170                          |
| NT 270Mb 8"          | IMSP                       | 518760                            | 259380                          |
| Fuj. 285Mb 14"       | IMSP                       | 545088                            | 272544                          |
| Fuj. 71Mb 8"         | EMSP                       | 131488                            | 65744                           |
| Fuj. 72Mb 14"        | EMSP                       | 135465                            | 67732                           |
| NEC 142Mb 8"         | EMSP                       | 261632                            | 130816                          |
| Fuj. 145Mb 8"        | EMSP                       | 263360                            | 131680                          |
| Fuj. 145Mb 14"       | EMSP                       | 270930                            | 135465                          |
| Fuj. 285Mb 14"       | EMSP                       | 539616                            | 269808                          |
| NEC 286Mb 8"         | EMSP                       | 541860                            | 270930                          |
| NEC 443Mb 9"         | EMSP                       | 835316                            | 417658                          |
| Fuj. 570Mb 14"       | EMSP                       | 1079232                           | 539616                          |
| Fuj. 586Mb 10.5"     | EMSP                       | 1108800                           | 554400                          |
| Fuj. 145Mb 8"        | AMSP                       | 269940                            | 134970                          |
| Fuj. 286Mb 8"        | AMSP                       | 548060                            | 274030                          |
| NEC 443Mb 9"         | AMSP                       | 849010                            | 424505                          |
| NEC 142Mb 8"         | AMSP                       | 252402                            | 126201                          |
| Fuj. 67Mb 5 ¼"       | MMSP                       | 141225                            | 70612                           |
| Fuj. 135Mb 5 ¼"      | MMSP                       | 282450                            | 141225                          |
| Max. 300Mb 5 ¼"      | MMSP                       | 507360                            | 253680                          |
| Fuj. 310Mb 5 ¼"      | MMSP                       | 637875                            | 318937                          |

---

The formula used in computing the total number of 512-byte sectors on your disk is in the section "Calculating Total Sectors Available" in the chapter *Configuring Disks*.

---

## How to Use This Program

You will first see the menu of options:

Please select desired option:

- f - Format the disk
- F - Format (No delay in listing bad sectors)
- s - Spare additional sectors on the disk
- r - Read the disk for media quality verification
- l - List the known bad sectors on the disk
- v - Verify sectors (EMSP disks, P/90, P/95)
- q - Quit

Type in your desired option. The program loops back to this menu after each option completes. If an error has been encountered, an appropriate error message will be displayed. If an error was encountered during formatting the disk or sparing sectors/tracks on the disk, you will be told that the disk is not reliable. You might merely need to run the option again, or you might have a hardware problem. The error shown indicates which situation you have.

After you select the desired option, you will be asked to indicate which type of controller you have.

If you indicate the wrong type, you will either get errors during the option process or get a bus error and have to reboot. You will next enter the physical disk number of the disk you will be working on. The IMSP supports disks 0 through 3; the EMSP, AMSP (P/95), and SCSI (P/90) support disks 0 through 15.

Following is a sample session from each option for each controller type. Prompts will be for string data, usually with a sample format given, numbers, usually with a minimum and maximum value shown, y or n responses to yes/no questions or other meaningful single characters.



---

## IMSP Examples

The following will be described:

- Format (with a bad track tape and without a bad track tape)
- Spare
- Read
- List

### Format (IMSP) Without a Bad Track Tape

Making a tape in this case is recommended.

Do you have a bad track tape to use for formatting? [y or n] **n**

Do you want to make one after the disk is formatted? [y or n] **y**

Please enter tape drive to use in format /dev/rrm/0m: **/dev/rrm/0m**

PLEASE INSERT YOUR BAD TRACK TAPE AND PRESS RETURN

Please enter the disk serial number: **23404**

The disk serial number is purely for documentation purposes (so you can be sure you have the right tape for a specific disk). No verification of the disk serial number is done.

- 1 - Drive Mfg. xx MB xx in.
- 2 - Drive Mfg. xxxMB xx in.
- 3 - Drive Mfg. xxx MB xx in.
- 4 - Drive Mfg. xxx MB xx in.
- 5 - Drive Mfg. xx MB x in.
- 6 - Drive Mfg. xxx MB xx.x in.
- 7 - Drive Mfg. xxx MB x in.
- x - Drive Mfg. xxx MB x in.
- 9 - Drive Mfg. xx MB x in.
- 10 - Drive Mfg. xx MB x in.

---

Disk type? [1-11]: 1

- 1 - Disk Serial Number: 23404
- 2 - Total Number of Cylinders: 823
- 3 - Number of Fixed Heads: 5
- 4 - Sectors per Track: 34
- 5 - Data Bytes Per Sector: 512
- 6 - Number of Alternate Cylinders: 20
- 7 - File System block size: 1024
- 8 - Default boot name: /unix

Number of line entry to change or <RETURN> <RETURN>

Defective tracks on fixed? (cyl,head) [0-822,0-4]: 4,1

Defective tracks on fixed? (cyl,head) [0-822,0-4]:

| cyl    | head   | spared to: | cyl        | head   |
|--------|--------|------------|------------|--------|
| 4(0x4) | 1(0x1) |            | 412(0x19c) | 4(0x4) |

formatting  
testing

Disk formatted and initialized successfully!

In the preceding example, we formatted a disk of a known type. It is possible to format other disks by selecting the OTHER type. You will be prompted for the necessary information. The data bytes per sector, number of alternate cylinders, file system block size and boot name are all default values. These can be altered by selecting the appropriate line number when prompted to do so. The altered values will be the ones written to your tape.

The cylinder and head of each track spared will be displayed with the alternate track to which it has been spared. To terminate input of the defective tracks, just press return at the prompt.

---

## With a Bad Track Tape

Do you have a bad track tape to use for  
formatting? [y or n] **y**

Please enter tape drive to use in  
format /dev/rrm/0m: **/dev/rrm/0m**

PLEASE INSERT YOUR BAD TRACK TAPE AND PRESS RETURN

- 1 - Disk Serial Number: 23404
- 2 - Total Number of Cylinders: 823
- 3 - Number of Fixed Heads: 5
- 4 - Sectors per Track: 34
- 5 - Data Bytes Per Sector: 512
- 6 - Number of Alternate Cylinders: 20
- 7 - File System block size: 1024
- 8 - Default boot name: /unix

Number of line entry to change or <RETURN> **<RETURN>**

Enter bad tracks from tape? [y or n] **y**

Bad track: cyl 4 head 1  
formatting  
testing

Disk formatted and initialized successfully!

In this example, the disk type information is read from tape. You can choose to alter the information by selecting the appropriate line number when prompted to do so. You will be given the option of writing the altered information to your tape.

You have the option of reading the defective track information from your tape or entering it by hand. If you choose the latter, you have the option of writing the new track information to your tape, unless you have altered your tape header information, in which case the hand-entered tracks will be written to tape automatically. The cylinder and head of each track spared will be displayed.

---

### **Spare (IMSP)**

Defective tracks on fixed? (cyl, head)

[0-<# cylinders>,0-<# heads>]: **3,1**

track 3/1 spared

Defective tracks on fixed? (cyl, head)

[0-<# cylinders>,0-<# heads>]:

Do you want to make a new track tape? [y or n] **y**

Disk serial number? **23030**

Please enter tape drive to use in format

/dev/rrm/0m: **/dev/rrm/0m**

PLEASE INSERT YOUR BAD TRACK TAPE AND PRESS RETURN

It is possible to make a new bad track tape whenever you spare tracks on an IMSP controlled disk, and it is highly recommended that you do so.

### **Read (IMSP)**

Disk reads complete

This option will only display information if bad tracks which are not spared are encountered.

### **List (IMSP)**

Output spared sector track list on tape? [y or n] **y**

Please enter tape drive to use in format

/dev/rrm/0m: **/dev/rrm/0m**

PLEASE INSERT YOUR BAD TRACK TAPE AND PRESS RETURN

| cyl    | head   | spared to: | cyl    | head |
|--------|--------|------------|--------|------|
| 4(0,4) | 1(0x1) | 412(0x19c) | 4(0x4) |      |

---

## EMSP Examples

This section contains the following:

- Format (with and without a bad track tape)
- Spare
- Read
- List
- Verify

### Format (EMSP) Without a Bad Track Tape

It is recommended that you make a tape at this time.

Do you have a bad track tape to use for formatting?

[y or n] **n**

Do you want to make one after the disk is formatted? [y or n] **y**

Please enter tape drive to use in format /dev/rrm/0m: **/dev/rpt/0m**

PLEASE INSERT YOUR BAD TRACK TAPE AND PRESS RETURN

Please enter the disk serial number: **20340**

- 1 - Fujitsu 72 MB 14 in.
- 2 - Fujitsu 145 MB 14 in.
- 3 - Fujitsu 285 MB 14 in.
- 4 - Fujitsu 570 MB 14 in.
- 5 - Fujitsu 71 MB 8 in.
- 6 - Fujitsu 586 MB 10.5 in.
- 7 - NEC 142 MB 8 in.
- 8 - NEC 443 MB 9 in.
- 9 - NEC 22 MB 8 in. IMSP only!
- 10 - NEC 36 MB 8 in. IMSP only!
- 11 - OTHER

Disk type? [1-1] **4**

- 1 - Disk Serial Number: 20340
- 2 - Total Number of Cylinders: 589

- 
- 3 - Number of Fixed Heads: 7
  - 4 - Sectors per Track: 34
  - 5 - Physical Bytes Per Sector: 590
  - 6 - Number of Alternate Sectors: 2
  - 7 - Default boot name: /unix
  - 8 - Number of patterns: 1
  - 9 - Number of retries: 1

Number of line entry to change or <RETURN> **<RETURN>**

Input bad sectors in Plexus format (cylinder, head and logical sector) or  
Manufacturer format (cylinder, head, offset and length)? [p or m] **p**

Defective sector (cyl, head, sec) [0-822, 0-4, 0-34]: **0,2,10**

Defective sector (cyl, head, sec) [0-822, 0-4, 0-34]: **3,1,20**

Defective sector (cyl, head, sec) [0-822, 0-4, 0-34]: **16,4,5**

Defective sector (cyl, head, sec) [0-822, 0-4, 0-34]:

Formatting disk, with 34 sectors/track  
Formatting disk, writing 65212 sectors  
Formatting disk, writing 65212 sectors  
Formatting disk, writing 9758 sectors  
retry(0)

Writing cylinder

10,20,....

Reading cylinder

10,20,....

Bad cylinder head sector (logical)

0 2 10

3 1 20

16 4 5

Formatting disk with 2 spare sectors/track  
Formatting disk, writing 65184 sectors  
Formatting disk, writing 65184 sectors  
Formatting disk, writing 1564 sectors

### **With a Bad Track Tape**

Do you have a bad track tape to use for  
formatting? [y or n] **y**

Please enter tape drive to use in

---

format /dev/rrm/0m: /dev/rrm/0m

PLEASE INSERT YOUR BAD TRACK TAPE AND PRESS RESET

- 1 - Disk Serial Number: **20340**
- 2 - Total Number of Cylinders: **589**
- 3 - Number of Fixed Heads: **7**
- 4 - Sectors per Track: **34**
- 5 - Physical Bytes Per Sector: **590**
- 6 - Number of Alternate Sectors: **2**
- 7 - Default boot name: **/unix**
- 8 - Number of patterns: **1**
- 9 - Number of retries: **1**

Number of line entry to change or <RETURN> **<RETURN>**

Enter bad tracks from tape? [y or n] **y**

Formatting disk with 34 sectors/track.  
Formatting disk, writing 65212 sectors.  
Formatting disk, writing 65212 sectors.  
Formatting disk, writing 9758 sectors.

retry(0)

writing cylinder

10,20,30,...

reading cylinder

10,20,30,....

Bad cylinder head sector (logical)

0 2 10

3 1 20

16 4 5

formatting disk with 2 spare sectors/track

formatting disk, writing 65184 sectors

formatting disk, writing 65184 sectors

formatting disk, writing 1564 sectors

In the first example, if you are formatting a disk of a type other than the one defined, you can select #11 and define the relevant information. To terminate the defective sector input, press <return> at the prompt.

---

In the second example, if you make changes to the tape header information, or elect to enter your defective sector list manually, you will be given the option of rewriting your bad track tape.

The program displays the lists of cylinders being written and read and the count of sectors being formatted on disk as an indication of its progress.

### **Spare (EMSP)**

Input bad sectors in Plexus format (cylinder, head and logical sector) or Manufacturer format (cylinder, head, offset and length)? [p or m] p

Defective sector (cyl, head, sec) [0-588,0-6,0-33]: 1,1,1

Defective sector (cyl, head, sec) [0-588,0-6,0-33]:

Bad cylinder head sector (logical)

1 1 1

DON'T FORGET TO MAKE A NEW BAD TRACK TAPE!!  
USE THE 'list' OPTION.

To terminate input of the defective sectors, press return at the prompt. Due to the program design, it is not possible to make a new track tape during running this option, as it is with an IMSP controller. Be sure to run the list option and make a new bad track tape, so that you always have a good tape.

### **Read (EMSP)**

Reading cylinder

10,20,30...

No spared sectors -or-

Bad cylinder head sector (logical)

# # #

This program will read the disk and notify you of any bad sectors it finds which need to be spared.

### **List (EMSP)**

Checking headers for cylinder:

0,



---

10,

Output spared sector list to tape? [y or n] **y**

Disk serial number?

Please enter tape drive to use in format /dev/rrm/0m: **/dev/rrm/0m**

PLEASE INSERT YOUR BAD TRACK TAPE AND PRESS RETURN

Press return to see the spared sector list:

Bad cylinder head sector (logical)

  #     #     #

You will not see the tape prompts unless you choose to make a tape.

### **Verify (EMSP)**

This test will invalidate data on the sector tested!

Continue? [y or n] **y**

Defective sector (cyl,head,sec) [0-x,0-y,0-z]:

Number of format patterns? [1-7]:

Number of format retries? [1-10]:

retry(#)

This will show a retry entry for each retry of each pattern. If the pattern written to the disk does not match the pattern read from the disk, an error will be indicated.

### **AMSP Examples**

This section contains the following:

- Format (with and without a bad track tape)
- Spare
- Read
- List

### **Format (AMSP) Without a Bad Track Tape**

It is recommended that you make a tape at this time.

Do you have a bad track tape to use for formatting?

[y or n] **n**

Do you want to make one after the disk is

---

formatted? [y or n] **y**

Please enter tape drive to use in  
format [/dev/rrm/0m]: **<c/r>**

PLEASE INSERT YOUR BAD TRACK TAPE AND PRESS RETURN

Please enter the disk serial number: **20340**

- 1 - NEC 167 MB 8 in.
- 2 - Fujitsu 337 MB 8 in.
- 3 - Fujitsu 168 MB 8 in.
- 4 - Fujitsu 520 MB 9 in.
- 5 - OTHER

Disk type? [1-1] **4**

- 1 - Disk Serial Number: 20340
- 2 - Total Number of Cylinders: 760
- 3 - Number of Fixed Heads: 19
- 4 - Sectors per Track: 60
- 5 - Physical Bytes Per Sector: 604
- 6 - Number of Alternate Tracks: 50
- 7 - Default boot name: /unix
- 8 - Number of patterns: 1
- 9 - Number of retries: 1
- 10 - Number of blocks in minor dev 0: 849010
- 11 - Gap 1: 12
- 12 - Gap 2: 12

Number of line entry to change or <RETURN> **<RETURN>**

Input bad sectors in Plexus format (cylinder, head and logical sector) or  
Manufacturer format (cylinder, head, offset and length)? [p or m] **p**

Defective sector (cyl, head, sec) [0-822, 0-4, 0-34]: **0,2,10**

Defective sector (cyl, head, sec) [0-822, 0-4, 0-34]: **3,1,20**

Defective sector (cyl, head, sec) [0-822, 0-4, 0-34]: **16,4,5**

Defective sector (cyl, head, sec) [0-822, 0-4, 0-34]:

Formatting disk, with 60 sectors/track

retry(0)

Writing cylinder

Reading cylinder

---

Bad cylinder head sector (logical)

|    |   |    |
|----|---|----|
| 0  | 2 | 10 |
| 3  | 1 | 20 |
| 16 | 4 | 5  |

Displays for each disk follow:

Disk type? [1-4] **1** (NEC 167Mb 8")

- 1 - Disk Serial Number: 12345
- 2 - Total Number of Cylinders: 1024
- 3 - Number of Fixed Heads: 8
- 4 - Sectors per Track: 32
- 5 - Physical Bytes Per Sector: 640
- 6 - Number of Alternate Tracks: 50
- 7 - Default boot name: /unix
- 8 - Number of patterns: 1
- 9 - Number of retries: 1
- 10 - Number of blocks in minor dev 0: 252402
- 11 - Gap 1: 16
- 12 - Gap 2: 16

---

Disk type? [1-4] **2** (Fuj 337Mb 8")

- 1 - Disk Serial Number: 54321
- 2 - Total Number of Cylinders: 832
- 3 - Number of Fixed Heads: 10
- 4 - Sectors per Track: 68
- 5 - Physical Bytes Per Sector: 594
- 6 - Number of Alternate Tracks: 50
- 7 - Default boot name: /unix
- 8 - Number of patterns: 1
- 9 - Number of retries: 1
- 10 - Number of blocks in minor dev 0: 849010
- 11 - Gap 1: 12
- 12 - Gap 2: 12

Disk type? [1-4] **3** (Fuj 168Mb 8")

- 1 - Disk Serial Number: 24680
- 2 - Total Number of Cylinders: 823
- 3 - Number of Fixed Heads: 10
- 4 - Sectors per Track: 34
- 5 - Physical Bytes Per Sector: 586
- 6 - Number of Alternate Tracks: 50
- 7 - Default boot name: /unix
- 8 - Number of patterns: 1
- 9 - Number of retries: 1
- 10 - Number of blocks in minor dev 0: 269940
- 11 - Gap 1: 8
- 12 - Gap 2: 8

### **With a Bad Track Tape**

Do you have a bad track tape to use for  
formatting? [y or n] **y**

Please enter tape drive to use in  
format [/dev/rrm/0m]: **<cr>**

PLEASE INSERT YOUR BAD TRACK TAPE AND PRESS RESET

- 
- 1 - Disk Serial Number: **20340**
  - 2 - Total Number of Cylinders: **760**
  - 3 - Number of Fixed Heads: **19**
  - 4 - Sectors per Track: **60**
  - 5 - Physical Bytes Per Sector: **604**
  - 6 - Number of Alternate Tracks: **50**
  - 7 - Default boot name: **/unix**
  - 8 - Number of patterns: **1**
  - 9 - Number of retries: **1**
  - 10 - Number of blocks in minor dev 0: **849010**
  - 11 - Gap 1: **12**
  - 12 - Gap 2: **12**

Number of line entry to change or <RETURN> **<RETURN>**

Enter bad tracks from tape? [y or n] **y**

Formatting disk with 60 sectors/track.

retry(0)

writing cylinder

reading cylinder

Bad cylinder head sector (logical)

|    |   |    |
|----|---|----|
| 0  | 2 | 10 |
| 3  | 1 | 20 |
| 16 | 4 | 5  |

To terminate the defective sector input, press <return> at the prompt.

In the second example, if you make changes to the tape header information, or elect to enter your defective sector list manually, you will be given the option of rewriting your bad track tape.

The program displays the lists of cylinders being written and read and the count of sectors being formatted on disk as an indication of its progress.

### **Spare (AMSP)**

Input bad sectors in Plexus format (cylinder, head and logical sector) or Manufacturer format (cylinder, head, offset and length)? [p or m] **p**

---

Defective sector (cyl, head, sec) [0-588,0-6,0-33]: **1,1,1**

Defective sector (cyl, head, sec) [0-588,0-6,0-33]:

Bad cylinder head sector (logical)

1 1 1

DON'T FORGET TO MAKE A NEW BAD TRACK TAPE!!

USE THE 'list' OPTION.

To terminate input of the defective sectors, press return at the prompt. Due to the program design, it is not possible to make a new track tape during running this option, as it is with an IMSP controller. Be sure to run the list option and make a new bad track tape, so that you always have a good tape.

#### Read (AMSP)

Reading cylinder

No spared sectors *-or-*

Bad cylinder head sector (logical)

# # #

This program will read the disk and notify you of any bad sectors it finds which need to be spared.

#### List (AMSP)

Checking headers for cylinder:

0,

10,

Output spared sector list to tape? [y or n] y

Disk serial number?

Please enter tape drive to use in

format [/dev/rrm/0m]: **<cr>**

PLEASE INSERT YOUR BAD TRACK TAPE AND PRESS RETURN

Press return to see the spared sector list:

Bad cylinder head sector (logical)

# # #

You will not see the tape prompts unless you choose to make a tape.

---

## P/90 Examples

Disks ordered with your P/90 system have been formatted by Plexus at the factory. DO NOT run **dformat** on a new P/90 system. If you need to reformat a disk or purchase an unformatted disk, use the **dformat** program to format the disk.

Formatting a P/90 disk is extremely simple. The **dformat** options available on the P/90 include:

- Format (with and without a bad track tape)
- Spare
- Read
- List
- Verify

The **dformat** menu for each of these options is the same for the P/95 and the P/90. After selecting the proper controller (SMD for P/95 or ESDI for the P/90), ensuring that the proper tape drive is selected (`/dev/rpt/0m`), and selecting the disk type from the menu, the **dformat** program lists a menu of technical information. See the section "AMSP Examples" earlier in this chapter and follow the menus in these examples. Press `<cr>` after each menu selection.

## mkfs

The stand-alone program, **mkfs**, like its `/etc/mkfs` counterpart, creates file systems. It declares that a certain range of disk addresses is to be considered a file system, and sets up disk blocks (free list, superblock, the i-list) within that file system for Sys5 to maintain housekeeping information about the files within the file system.

The controlling blocks contain information such as how many files are in the file system, how many blocks are free, where files live on the disk, and how big they are.

When creating file systems to be mounted (i.e., file systems other than the root file system), you can run `/etc/mkfs` under Sys5 or

---

**mkfs** of the stand-alone environment. When creating the root file system, however, stand-alone **mkfs** *must* be used because it prepares a file system so Sys5 and its utilities (the root file system) can be **restored** from tape onto the disk.

[ **NOTE** ] Plexus has already run stand-alone **mkfs** on the root file system. You do not will not need to run stand-alone **mkfs** unless you alter the default size or location of the root file system or the swap area.

Because **mkfs** re-initializes the superblock and i-lists, any data that is already on the logical disk, though not overwritten, becomes inaccessible, since Sys5 loses all pointers to it. Therefore, **mkfs** must be used with caution **AND ONLY AFTER A BACKUP!**

After running **mkfs** and before **mounting** the file system you made, it is a good idea to run **fsck** on the new file system, to insure the integrity of the file system.

To create a file system using stand-alone **mkfs** from disk, enter at the boot ( : ) prompt:

```
/stand/mkfs
```

The system returns:

```
$$ mkfs
```

Next enter:

```
/dev/dsk/Xsx xxxx m n
```

where:

*X* is the physical disk number ( 0 - n)

*x* is the logical disk number ( 0 - 15)

*xxxx* is the file system size in 1K blocks

*m* is the software interleave

*n* is blocks/cylinder

See the following table for actual values of *m* and *n*.



**Table 15-2. Software Interleave and Blocks per Cylinder**

| <u>Drive Type</u> | <u>Cont</u> | <u>Sect/ Cyl</u> | <u>cylsize<br/>Blocks/<br/>Cyl</u> | <u>softint<br/>Software<br/>Interleave</u> |
|-------------------|-------------|------------------|------------------------------------|--------------------------------------------|
| NEC 22Mb 8"       | IMSP        | 102              | 51                                 | 1                                          |
| NEC 36Mb 8"       | IMSP        | 170              | 85                                 | 1                                          |
| Fuj. 67Mb 5 ¼"    | MMSP        | 175              | 88                                 | 9                                          |
| Fuj. 71Mb 8"      | IMSP        | 238              | 119                                | 1                                          |
| Fuj. 71Mb 8"      | EMSP        | 224              | 112                                | 7*                                         |
| Fuj. 72Mb 14"     | IMSP        | 170              | 85                                 | 1                                          |
| Fuj. 72Mb 14"     | EMSP        | 165              | 82                                 | 7*                                         |
| Fuj. 145Mb 8"     | AMSP        | 330              | 165                                | 7*                                         |
| Fuj. 135Mb 5 ¼"   | MMSP        | 350              | 175                                | 9                                          |
| NEC 142Mb 8"      | IMSP        | 272              | 136                                | 1                                          |
| NEC 142Mb 8"      | EMSP        | 256              | 128                                | 7*                                         |
| NEC 142Mb 8"      | AMSP        | 248              | 124                                | 7*                                         |
| Fuj. 145Mb 8"     | IMSP        | 340              | 170                                | 1                                          |
| Fuj. 145Mb 8"     | EMSP        | 320              | 160                                | 7*                                         |
| Fuj. 145Mb 14"    | IMSP        | 340              | 170                                | 1                                          |
| Fuj. 145Mb 14"    | EMSP        | 330              | 165                                | 7*                                         |
| Max. 300Mb 5 ¼"   | MMSP        | 420              | 210                                | 9                                          |
| NT 270Mb 8"       | IMSP        | 396              | 198                                | 1                                          |
| Fuj. 286Mb 8"     | AMSP        | 670              | 335                                | 7*                                         |
| Fuj. 285Mb 14"    | IMSP        | 544              | 272                                | 1                                          |
| Fuj. 285Mb 14"    | EMSP        | 528              | 264                                | 7*                                         |
| NEC 286Mb 8"      | EMSP        | 660              | 330                                | 7*                                         |
| Max 310Mb 5 ¼"    | MMSP        | 525              | 262                                | 9                                          |
| NEC 443Mb 9"      | EMSP        | 1102             | 551                                | 7*                                         |
| NEC 443Mb 9"      | AMSP        | 1120             | 560                                | 7*                                         |
| Fuj. 570Mb 14"    | EMSP        | 1056             | 528                                | 12                                         |
| Fuj. 586Mb 10.5"  | EMSP        | 1320             | 660                                | 7*                                         |

\* **softint** is optimal for standard Sys5 operation. **softint** can be changed to be optimal for your normal operation.

See the section "Cylinder Size" in the chapter *Configuring Disks* for equations describing how to compute **mkfs** cylinder size.

---

## **dconfig**

**dconfig** is a rather comprehensive program that determines several disk and system factors. You can run **dconfig** to obtain or change the status of any of the following parameters:

- Disk identification (**id**)
- Operating system kernel name (primary bootname)
- Alternate operating system kernel name (secondary bootname)
- Disk information specific to the actual disk unit installed, including number of cylinders, sector size, sectors per track, etc.
- System's node name (if you want to give your system a name to put it on a **uucp** network)
- Which logical disk will contain the root file system and perform pipes and **dumps**
- Which logical disk will provide the swap area
- Address and size of the swap area
- Boundaries and default start addresses of each potential logical disk

**[ NOTE ]** These parameters are vital to the system. Some parameters (such as disk cylinders, sector size, etc.) have no alternate values for your system. Changing them simply disables your system. Other parameters can be changed if done carefully and consistently, such as: device numbers for root, swap, etc., swap size and location, start addresses of the logical disks.

Changing your nodename will not affect internal system performance, but other nodes on your **uucp** net must be notified if you still want their messages to find your system.

You can run **/etc/dconfig** or stand-alone **dconfig** to look at the status of any of the parameters previously mentioned. You should

---

run stand-alone **dconfig**, however, to change any of the parameters. You will most likely invoke stand-alone **dconfig** to perform the following functions:

1. Create a Sys5 node name for **uucp** (see the chapter *UUCP Administration* in the *Sys5 UNIX Administrator's Guide*).
2. Enlarge or relocate the swap area (see the chapter *Changing Root and Swap* in this manual).
3. Change the default organization of logical file systems (see the chapter *Configuring Disks* in this manual).

For additional information describing how to use stand-alone **dconfig** see the chapters *Configuring Disks* and *Changing Root and Swap* in this manual.

## **fsck**

**fsck** is a file system consistency checker. To run stand-alone **fsck** from disk at the boot prompt ( : ) enter:

```
/stand/fsck
```

The system returns:

```
$$ fsck
```

At the **\$\$ fsck** prompt, press **<cr>** to check the file systems listed in **/etc/checklist**. See the section "Step-by-Step Disk Configuration Overview" in the chapter *Configuring Disks* for additional information on editing **/etc/checklist**. To check specific file systems, enter the disk device name at the **\$\$ fsck** prompt (user response in bold). Enter:

```
$$ fsck <cr> (to check file systems in /etc/checklist)
```

OR

```
$$ fsck /dev/dsk/Xsxx (to check a specific file system(s)).
```

where:

**X** is the physical disk number

**xx** is the logical disk number (0-15)

---

For further information, see the the document **fsck** in the *Plexus Sys5 UNIX Administrator Guide*. This describes in great detail what **fsck** does and how it works. Stand-alone **fsck** can check a maximum of five file systems per invocation.

---

## restor

The stand-alone program **restor** is the partner of **/etc/dump**. **restor** puts the contents of **dump** tapes back onto disk. Use stand-alone **restor** to restore the the root file system from **dump** tape. Stand-alone **restor** and **/etc/restor** are documented in the section "Using restor" in the chapter *Backup Procedures*.

## dd

This program works as described in the manual pages. See the chapter, *Backup Procedures*, for an explanation of how **dd** fits in with the other Sys5 programs that do tape I/O.

## fbackup

**fbackup** is a fast, stand-alone Plexus program that copies from disk to tape, or tape to disk, in streaming mode, blocking files at 32 disk sectors per tape record. **fbackup** is not available on VMEbus-based systems.

See the chapter *Backup Procedures* for an explanation of how **fbackup** fits in with the other Sys5 programs that perform tape I/O.

**fbackup** is faster than **/etc/dump** and writes in a format that is understood by **dd**. You should use **fbackup** rather than **/etc/dump** if you need the speed.

To use **fbackup**, you need to know:

1. the starting disk address of the file system (obtained from **dconfig**)
2. its length in 512 byte disk sectors

---

Multiply the number of 1K blocks you specified when you executed **mkfs** by 2 to obtain 512-byte sectors. Use the number of 512-byte sectors.

Your file systems cannot in any case be larger than the logical disk sizes reported by **dconfig**, and **fbackup** does not mind if the file system size figure is larger than the size of the actual file system. If you have changed the file system sizes to larger than standard values without having increased the default logical disk sizes, you will have trouble.

**fbackup** does not care what you back up and does not recognize file system boundaries. It just backs up the sectors you specify (i.e. copies to tape). You should care what gets put on the tape and be sure you know what the boundaries are (in case you have to recover data).

## Other Stand-alone Programs

Plexus provides stand-alone versions of the following commonly used Sys5 commands. These stand-alone programs are described in the *Plexus Sys5 UNIX User's Reference Manual* and the *Plexus Sys5 UNIX Administrator Guide*.

- fsdb**    File system debugger.
- od**      Octal dump. It allows dumping of a file in various octal, decimal, or hex formats. It is mainly a debugging tool.
- ls**      List files and directories.
- cat**     Concatenate, used to display files.
- du**      Reports disk usage.

C

C

C

## INDEX

### A

- accounting, 5-17
- ACP, 2-28
- ACP dump, how to perform, 14-16
- ACP I/O port connections, 6-8
- ACP LED indicators, 14-16
- AMSP disk configuration values, 8-20
- adding a new user, 10-3
- additional file systems, creating, 8-7
- administrator, alternate, 1-6
- administrator, duties of, 1-4
- advanced communications processor, 2-28
- Arbiter board, P/95, 2-21
- auto boot, 5-9

### B

- backing up file systems, 9-10, 11-7
- backing up files, 11-19
- backplane, 2-23
- backup program, very fast, 15-40
- backup programs, 11-1
- backup script, 11-22
- backups, automating, 11-21
- backups, file systems, 11-7
- backups, files, 11-19
- backups, scheduling, 11-10
- backups, when to make, 11-6
- bcheckrc file, 3-22, 5-10
- /bck directory, 3-6
- /bin directory, 3-8
- blocks available, total 1K, 8-8
- blocks per cylinder, 8-32, 9-15, 15-38
- blocks, indirect, 7-7
- boot block, 7-6
- booting the system, 4-3
- brc file, 3-22
- bytes per sector (EMSP), 8-11

### C

- cannot fork, 13-18
- cartridge tape and disk controllers, 2-30
- cartridge tape drive, 5-19
- cat, standalone, 3-31
- CCB, 2-12
- central processing unit, 2-24
- changing multiple group affiliation, 10-14
- checking file systems, 14-4, 15-40



## INDEX

- checklist file, 3-22
- clear to send signals, 6-10
- commands, system administration, 3-22
- common circuits board, 2-9
- computer log, 1-8
- conf.c, 3-18
- configuring a second drive, 8-30
- configuring disks, 8-1, 8-14
- connecting a modem, 6-28
- connecting printers, 6-18, 6-22
- console pinout table, 4-2
- console, connecting, 4-2
- console, how to unfreeze, 13-16
- controller, disk (AMSP), 2-32
- controller, disk (EMSP), 2-32
- controller, disk (ESDI), 2-33
- controller, disk and cartridge tape (IMSP), 2-30
- controllers, disk and tape, 2-30
- copying files, 7-11
- cp command, 7-11
- cpio, 11-2, 11-4, 11-19, 11-20
- CPU, 2-24
- creating additional file systems, 8-6, 8-14
- creating devices in /dev, 6-2, 8-5
- creating file systems, 8-6, 15-36
- creating group entries, 10-6
- creating login directories, 10-3
- creating lost+found, 8-28
- creating password entries, 10-4
- creating user accounts, 10-1
- crisis recovery, 14-1
- cron, 3-22
- cshprofile, 3-23
- .cshrc file, 10-8
- CTS signals, 6-10
- cylinder size, 8-12
- cylinder, blocks per, 8-32, 9-15
- cylinder, sectors per, 8-32
- cylinders, number of, 8-10, 8-11

## INDEX

### D

- data area, 7-7
- data carrier detect signals, 6-10
- data, how to recover, 14-5
- DCD signals, 6-10
- dconfig cylinder size, 8-12
- dconfig, 3-31, 8-12, 9-7, 9-11, 9-21, 9-29
- dconfig, standalone, 3-31
- dd, 11-2, 11-4, 11-26
- dd, standalone, 3-31
- ddate, 3-23
- /dev directory, 3-8
- device files, 3-8
- device names, cartridge tape drives, 5-19
- device names, reel-to-reel tape drives, 5-23
- device numbers, major, 3-10 thru 3-17
- device numbers, minor, 8-25
- devices, creating in /dev, 6-2, 8-5
- devices, peripheral, 2-38
- df command, 7-19
- dformat, standalone, 3-31
- dial in modem, setting up, 6-30
- directories, restoring, 14-6
- directory ownership, how to change, 10-10
- disk and cartridge tape controller (IMSP), 2-30
- disk configuration values, 8-11, 8-19, 8-20
- disk configuration, 8-1, 8-14, 15-37
- disk configuration, second drive, 8-30
- disk controller (EMSP), 2-32
- disk controller (AMSP), 2-32
- disk controllers, 2-30
- disk drives, 2-34
- disk free (df) command, 7-19
- disk free space, determining, 7-19
- disk mirroring, 12-1
- disk organization, 7-2, 8-15
- disk, optical, 2-34
- disks, how to mirror, 12-1
- disks, logical, 7-2
- disks, physical, 7-2
- drives, disk, 2-34
- drives, tape, 2-35
- du, standalone, 3-31
- dump, 3-23, 11-3, 11-4, 11-6, 11-10
- dumps, full, 11-11
- dumps, incremental, 11-13

## INDEX

### E

- eccdaemon, 3-23
- EMSP disk configuration values, 8-19
- error messages, cannot fork, 9-4, 13-18
- error messages, no more processes, 9-4, 13-18
- error messages, reel-to-reel tape drives, 5-32, 5-35
- error messages, swap space too small, 9-4
- error messages, Sys5: panic: out of swap space, 9-4
- error messages, umount, 7-17
- /etc directory, 3-22
- /etc/dump, 11-3, 11-4, 11-6, 11-10
- /etc/getty, 13-5
- /etc/gettydefs, 13-7
- /etc/init, 4-10
- /etc/inittab, 4-10
- /etc/labelit, 11-25
- /etc/mirutil, 12-1, 12-4
- /etc/mnttab file, 7-17
- /etc/rc file, 3-25, 4-12 thru 4-32
- /etc/restor, 11-4, 11-14
- /etc/shutdown, 4-8
- /etc/TIMEZONE file, 4-22
- /etc/ttyconf, 6-2
- executable files, 3-8
- exporting variables, 10-11

### F

- fast backup program, 15-40
- fbackup, 11-2, 11-3, 11-5, 11-7, 15-40
- fbackup, standalone, 3-31, 11-7
- file naming, 7-9
- file ownership, how to change, 10-12
- file size, maximum, 5-14
- file system checking, 14-4, 15-38
- file system, general information, 7-1
- file system, restoring, 14-9
- file systems, creating, 8-1, 8-6, 15-34
- file systems, how to back up, 7-14, 8-17
- file systems, how to repair, 14-4
- file systems, mounting, 7-15
- file systems, set up at Plexus, 7-13
- file systems, unmounting, 7-15
- files, backing up, 11-19
- files, copying, 7-11
- files, devices, 3-8
- files, executable, 3-8
- files, how to name, 7-9
- files, linking, 7-10

## INDEX

- files, moving, 7-12
- files, restoring, 11-16, 14-6
- files, types of, 3-4
- finc, 11-2, 11-3, 11-4, 11-25
- find, 11-19, 11-25
- forking, 13-10
- formatting program for hard disks, 15-14
- frec, 11-2, 11-3, 11-5, 11-25
- free space, 7-8
- freeing up frozen terminals, 13-13
- freeing up the system console, 13-16
- frozen terminals, 13-13
- fsck, 3-23, 4-5, 4-9, 14-4
- fsck, standalone, 3-31
- fsdb, 3-23
- fsdb, standalone, 3-31

### G

- getty, 3-23, 13-5
- gettydefs file, 3-23, 13-7
- group affiliation (multiple), changing, 10-14
- group entries, creating, 10-6
- group file, 3-23
- group ownership, how to change, 10-12
- grpck, 3-23

### H

- hard disks, formatting program, 15-14
- hardware components, 2-1
- hardware, P/35, 2-2
- hardware, P/55, 2-5
- hardware, P/60, 2-7
- hardware, P/75, 2-10
- hardware, P/90, 2-14
- hardware, P/95, 2-17
- help, standalone, 3-31
- holidays, 5-18

### I

- i-list, 7-6
- i-node number, 11-17
- ICP, 2-28
- ICP dump procedure, 14-12
- ICP I/O port connections, 6-5
- IMSP controller, 2-31
- IMSP disk configuration values, 8-19
- indirect blocks, 7-7
- init, 3-23, 4-11

## INDEX

- init 2, 4-11
- init a, 5-8
- init b, 5-8
- init c, 5-8
- init Q, 5-8
- init q, 5-8
- inittab, 3-24, 4-11
- input/output devices, how to set up, 6-1
- intelligent communications processor, 2-28
- interleave, 8-28, 9-15, 15-36
- introduction, 1-2

### K

- kernel, 3-2, 3-3, 15-13
- kernel, configuration file, 3-18
- kill command, 13-12
- killing system processes, 13-12

### L

- labelit, 3-24, 11-25
- Link cards, 2-21, 2-33
- linking files, 7-10
- ln command, 7-10
- loading tapes manually, reel-to-reel drives, 5-29
- loading tapes, reel-to-reel drives, 5-28
- log out, how to, 4-7
- logbook, 1-7
- logging in as root, 4-6
- logging out, how to, 4-7
- logical disks, 7-2
- login directories, creating, 10-3
- login incorrect message appears on terminal, 10-18
- login procedure, 4-3
- lost data, how to recover, 14-5
- lost+found directory, 3-6, 7-16
- lost+found, creating, 8-28
- ls, standalone, 3-31

### M

- MMSP disk configuration values, 8-20
- major device numbers, 3-9
- managing system processes, 13-1
- memory boards, 2-25
- memory dump procedure, 14-12, 14-16, 14-21
- memory dump, how to perform, 14-12, 14-16, 14-21
- microprocessor, 2-24
- minor device numbers, 8-25
- mirror disk, 12-1, 12-4

## INDEX

mirutil, 12-1, 12-4  
mkfs cylinder size, 8-27  
mkfs, 3-24, 8-4  
mkfs, standalone, 3-31  
mknod, 3-24, 8-5  
mknod, minor device numbers, 8-25  
/mnt directory, 3-6  
mnttab file, 7-17  
modem, setting up, 6-28  
modem, pinout table, 6-6, 6-9, 6-13  
modems, 2-38  
modes, system, 5-6  
motd, 3-24  
mount command, 3-24, 4-24, 8-27, 8-28  
mounting file systems, 7-15  
moving files, 7-12  
multi-user mode, 4-11, 5-7  
mv command, 7-12  
mmdir, 3-24

### N

ncheck, 3-24  
new user, how to add, 10-3  
no directory message appears on terminal, 10-18  
no login appears on terminal, 10-17  
no more processes, 13-18  
number of cylinders, 8-10, 8-11

### O

od, standalone, 3-31  
openup, 3-24  
operating system, overview, 3-2  
optical disk, 2-34

### P

P/35, hardware, 2-2  
P/55, hardware, 2-5  
P/60, hardware, 2-7  
P/75, hardware, 2-10  
P/90, hardware, 2-14  
P/95, hardware, 2-17  
parallel port connections, 6-7  
parallel printer pinout table, 6-7, 6-14  
parallel printer, connecting, 6-18  
passwd, 3-25  
passwd (bin), how to set up, 4-7  
passwd (root), how to set up, 4-7  
password entries, creating, 10-4

## INDEX

- passwords (system), how to set up, 4-6
- PATH, 10-10
- peripheral devices, 2-38
- physical disks, 7-2
- Plexus systems, introduction, 1-2
- power failure, recovering from, 14-1
- power supply, 2-24
- power up procedure, 4-3
- prime time, 5-18
- printer (parallel) pinout table, 6-7, 6-11, 6-14
- printer (parallel), connecting, 6-18
- printer (serial) pinout table, 6-5, 6-8, 6-13
- printer (serial), connecting, 6-22
- printers, 2-39
- process status command, 13-3
- processes, killing, 13-12
- processes, managing, 13-1
- .profile, 3-25
- .profile file, creating, 10-8
- .profile, initializing changes, 10-11
- programs, standalone, 15-1
- ps command, 13-3
- ps\_data, 3-25
- pwck, 3-25

## R

- rc file, 3-25, 4-12 thru 4-32
- rebooting the system, 4-3
- recovering lost data, 14-5
- reel-to-reel tape controller, 2-33
- reel-to-reel tape drive loading/unloading, 5-28
- reel-to-reel tape drive manual loading/unloading, 5-29
- reel-to-reel tape drive performance, 5-27
- reel-to-reel tape drive, device names, 5-23
- reel-to-reel tape drive, how to use, 5-19
- reel-to-reel tape drive, modes, 5-27
- reel-to-reel tape drives, error messages, 5-32, 5-35, 5-37
- repairing file systems, 14-4
- resetting the system, 4-3, 8-18
- response time, slow, 13-17
- restor, 3-25, 11-4, 11-14, 11-16
- restor, standalone, 3-31, 11-16
- restore from dump tape, 11-14, 15-28
- restoring directories, 14-6
- restoring file systems, 11-4, 11-14, 11-16, 14-6
- restoring files, 11-16, 14-6
- restoring the root file system, 11-14
- root directory, listing, 3-5

## INDEX

- root file system and swap area, increasing the size of, 9-9
- root file system, description, 3-5
- root file system, increasing the size of, 9-9
- root file system, organization, 3-5, 7-4
- root file system, restoring, 11-14
- root file system, subdirectories, 3-6
- root, how to become, 1-9
- root, powers of, 1-10
- root, precautions, 1-11
- root's login environment, setting up, 5-2

### S

- SAFE switch, 5-21
- scheduling dumps 11-10
- second drive, configuring, 8-34
- sectors available, total 512-byte, 8-8
- sectors available, calculating, 8-9
- sectors per cylinder, 8-32
- sectors per track, 8-11
- serial port connections, 6-5, 6-8, 6-12
- serial ports, 2-27
- serial printer pinout table, 6-6, 6-8, 6-13
- serial printer, connecting, 6-22
- setmnt, 3-25
- setting the terminal type, 10-15
- setting up printers, 6-18, 6-22
- setting up a modem, 6-28
- setting up I/O devices, 6-1
- setting up input/output devices, 6-1
- setting up root's login environment, 5-2
- setting up system passwords, 4-6
- setting up terminals, 6-3
- setting up ttys, 6-3
- shell, 3-3
- shutdown, 3-26, 4-8
- shutdowns, recovering from ungraceful, 14-3
- shutting down the system, 4-3, 4-8
- single-user mode, 5-6
- single-user mode, entering 4-10
- skip command, 15-12
- slow system, 13-17
- software components, 3-2
- software interleave, 8-28, 9-15, 15-36
- spawning, 13-10
- /stand directory, 3-7, 3-29
- /stand, programs residing in, 3-31
- standalone fbackup 11-5
- standalone programs, 15-1



## INDEX

- standalone programs, arguments to, 15-10
- standalone programs, cat, 15-41
- standalone programs, dconfig, 15-37
- standalone programs, dd, 15-40
- standalone programs, description of, 15-13
- standalone programs, dformat, 15-14
- standalone programs, du, 15-41
- standalone programs, fbackup, 15-40
- standalone programs, fsck, 15-38
- standalone programs, fsdb, 15-41
- standalone programs, general information, 15-3
- standalone programs, help program, 15-13
- standalone programs, how to load, 15-7
- standalone programs, how to run, 15-6
- standalone programs, ls, 15-41
- standalone programs, mkfs, 15-34
- standalone programs, od, 15-41
- standalone programs, restor, 15-40
- standalone programs, unix, 15-13
- standalone programs, use of, 3-29
- standalone programs, when to run, 3-29
- standalone restor, 11-15
- startup and shutdown, 4-3
- states, changing, 5-6
- states, system, 4-10
- states, user-definable, 5-6
- super user, 1-9
- superblock, 7-6
- swap area too small error messages, 9-4
- swap area, decreasing the size of, 9-5
- swap area, determining the size of, 9-3
- swap area, increasing the size of, 9-5, 9-9
- swap area, moving the location of, 9-5
- Sys5 overview, 3-2
- system administration commands, 3-22
- system administrator, alternate, 1-6
- system administrator, duties of, 1-4
- system console pinout table, 4-2
- system console, connecting, 4-2
- system console, how to unfreeze, 13-16
- system log, 1-8
- system modes, 5-6
- system process management, 13-1
- system processes, killing, 13-12
- system processes, managing, 13-1
- system shutdown, how to, 4-8
- system slowdown, 13-17
- system states, 4-10, 5-6

## INDEX

### T

- tape (cartridge) and disk controller (IMSP), 2-30
- tape backup procedure, 11-1
- tape backup script 11-22
- tape controllers, 2-30
- tape drive error messages, reel-to-reel drives, 5-32, 5-35, 5-37
- tape drive loading, reel-to-reel, 5-28
- tape drive manual loading, reel-to-reel, 5-29
- tape drive modes, reel-to-reel, 5-27
- tape drive performance, reel-to-reel, 5-27
- tape drive unloading, reel-to-reel, 5-28
- tape drive, cartridge, 5-19
- tape drives, 2-35
- tape drives, how to use, 5-19
- tape dumps, full, 11-11
- tape dumps, scheduling, 11-10
- tape utilities 11-2
- tar, 11-2, 11-4, 11-23
- terminal pinout table, 6-5, 6-8, 6-13
- terminal type, setting, 10-15
- terminals, 2-38
- terminals, how to unfreeze, 13-13
- terminals, setting up, 6-3
- thrashing, 13-17
- time, prime, 5-18
- TIMEZONE file, 4-22
- timezone information, 4-22
- /tmp directory, 3-7
- total blocks available, 8-8
- total sectors available, 8-8
- total sectors available, calculating, 8-9
- trouble shooting guide, 14-2
- troubleshooting guide, user accounts, 10-17
- ttyconf, 6-2
- ttys, setting up, 6-3
- ttytype, 3-26

### U

- umount, 3-26, 4-23
- umount error messages, 7-17
- umount example, 8-27
- ungraceful shutdown, recovering from, 14-3
- /unix file, 3-7
- unix overview, 3-2
- unloading tapes, reel-to-reel drives, 5-29

## INDEX

unmounting file systems, 4,23 7-15  
user accounts, creating, 10-1  
user accounts, troubleshooting guide, 10-17  
users, types of, 1-10  
/usr directory, 3-27  
/usr subdirectories, 3-27, 3-28  
utilities, 3-2  
utmp, 3-26

### V

variables, exporting, 10-8  
VCP, 2-29  
VCP, how to configure, 6-15  
VMEbus Communications Processor, 2-29  
VMEbus Communicatons Processor, how to configure, 6-15  
volcopy, 11-2, 11-3, 11-5, 11-8

### W

wall, 3-36  
who command, 13-8

## Reader Comment Form

There's always room for improvement, and we at Plexus Computers are interested in any comments and suggestions that would make this manual better. If you take the time to make comments, we'll take the time to read and consider them for the next revision. Mail your comments to us at this address:

Technical Publications Department  
Plexus Computers, Inc.  
3833 North First St.  
San Jose, CA 95134

Manual Name: *Sys5 Administrator's Handbook*

Publication Number: *98-05133.2 Ver. B*

1. Is there information that you think should be included (or removed)?

---

---

---

2. If you found errors, please list the page number and describe the error.

---

---

---

3. Any other comments?

---

---

Your Name (optional):

Company Name and Address:  
(or attach a business card)

