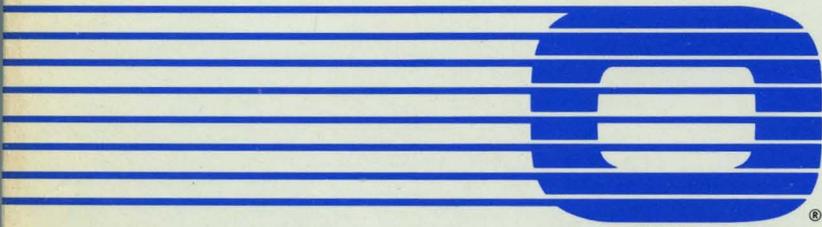


Volume 4
Operating Systems



OSBORNETM
COMPUTER CORPORATION

VOLUME 4
**Operating
Systems**

OSBORNE
EXECUTIVE
GUIDES

Part 1, *CP/M Plus*, was written by Mike Iannamico and Joseph Caggiano.

Part 2, *p-System*, was prepared by Heather Allen with technical assistance from Tom Swegles and Stan Stringfellow.

Part No. 3F00196-00

All rights to this publication are reserved. No part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or in any other manner, without prior permission from OSBORNE COMPUTER CORPORATION, 26538 Danti Court, Hayward, CA 94545.

Copyright © 1983 Osborne Computer Corporation, 26538 Danti Court, Hayward, CA 94545.

Osborne Executive is a registered trademark of Osborne Computer Corporation. WordStar®, copyright © 1981, and MailMerge® are registered trademarks of MicroPro International Corporation. SuperScript SuperCalc® and SuperData Interchange are trademarks of Sorcim Corporation. CP/M Plus™ is a trademark of Digital Research Corp. Microsoft BASIC, copyright © 1977-1981, by Microsoft. CBASIC, copyright © by Compiler Software, Inc. UCSD p-System is a trademark of the University of California. Personal Pearl™ is a trademark of Pearlssoft.

Osborne Computer Corporation reserves the right to make changes or improvements to the equipment, software, and documentation herein described at any time and without notice.

Notice: Please be advised that any tampering with the internal hardware of the Osborne Executive Computer may void the warranty.

FCC Notice

This equipment generates and uses radio frequency energy. If not installed and used properly, that is, strictly according to the manufacturer's instructions, the equipment may cause interference with radio and television reception.

This equipment has been type tested and found to comply with the limits for a **Class B** computing device as specified in FCC Rules, Part 15, Subpart J, which is designed to provide reasonable protection against such interference in a residential installation.

If the equipment does cause interference to radio or television reception, which can be determined by turning it ON and OFF, try to correct the interference by doing one or more of the following:

- Reorient the receiving antenna.
- the computer with respect to the receiver.
- Move the computer away from the receiver.
- Plug the computer into a different outlet so the computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful:

“How to Identify and Resolve Radio-TV Interference Problems”.

This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, Stock No. 004-000-00345-4.

To maintain Class B compliance use properly shielded and grounded cables when connecting this equipment to peripheral devices.

Warning

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

Contents

Part 1 1

CP/M Plus

Introduction 2

CP/M Plus Utilities 5

What are Utilities? 5

Section 1: Startup Utilities 7

The HELP Command 9

The DATE Command 11

Set the Clock 11

Time Display on Screen 13

Time Stamping Files 14

INITDIR: Preparing Space for the Time Stamp 14

SET: Recording Date and Time 15

SETDEF: Finding and Displaying Files 18

SETDEF Command Summary 20

Section 2: Diskette Attributes 21

Naming a Diskette 21

Making a Diskette Read Only 22

Passwords 23

Diskette Passwords 24

File Passwords 25

Password Options 26

Access to SYSTEM Files 27

SET Command Summary 28

SHOW and DIR: Checking Diskette Contents 28

Displaying A Diskette's Name 29

Displaying User Numbers	30
Detailed File Directories	31
Directory Options	32

Section 3: File Handling with PIP **34**

Copying Individual Files	35
Copying Several Files	35
Copying Groups of Files	36
Merging Files	37
PIP Options	38
List of PIP Options	38
Copying To and From External Devices	40
PIP External Device Options	41
PIP Summary	42

Section 4: Communicating with External Devices **43**

Physical and Logical Devices	43
Displaying Device Assignment	454
Assigning Physical Devices	47
Displaying Device Information	50
Baud Rates and XON/XOFF Protocol	51

Section 5: ED, The CP/M Plus Line Editor **52**

Creating a File	52
Editing Commands	53

Section 6: File Processing and Data Handling **56**

File Processing with SUBMIT	56
Multiple Submit Files	58
The PUT Command	59

PUT Command Options	60
The GET Command	62

Section 7: Running Assembly-Language Programs **64**

MAC and RMAC Assemblers	64
Assembling A Program	65
RMAC Command Options	67
Creating a Reference File	68
Displaying Relocatable Object Code	70
Preparing a File for Execution	71
Running A Program	72
Modifying Programs with SID	72

Part 2 **77**

p-System

Introduction **78**

Section 1 Tutorial **83**

Learning to use the p-System **85**

The p-System Operating System	81
The Development System	81
What's Included with Your p-System	82
Organization of This Manual	84

Preparing to Use The p-System **85**

Copying your Master p-System Diskette	85
Loading the p-System	85
The p-System's Promptline	86

Starting a Session and Preparing a Blank Diskette **90**

Setting the Date 90
 Formatting a Blank Diskette 92
 Creating a New Directory 94

Understanding File Names, File Types, and Volumes **97**

File Names 97
 File Types 98
 Volumes 100
 Putting it All Together 103

File Handling **104**

The Filer 104
 Leaving the Filer 105
 Creating an Empty File 105
 Wild Card Conventions 107
 Looking at the Directory 108
 Displaying the Extended Directory 110
 Changing File Names 111
 Removing a File from a Diskette 111
 Listing your Volume Names 112
 Copying Files 113
 Copying a Single File 113
 To Copy a File on the Same Drive 114
 To Copy Several Files 114
 To Copy an Entire Diskette 115
 To Copy to the Screen or Printer 116
 An Easy Way to Refer to Volumes 117
 Work Files 118

Handling Diskette Problems **121**

- Finding Bad Blocks 121
- Examining the Bad Block Area 122
- Rearranging Files on the Diskette 124
- Recovering Erased Files 125
- Another Way to Recover Lost Files 127
- Preventing Directory Trouble 127

Some Useful Commands **131**

- Reinitializing the System 131
- Rerunning the Last Program 131
- Leaving the p-System 132
- Saving Your Commands in a File 132

More about Diskette Preparation and Handling **135**

- Putting Boot Tracks on a Diskette 136
- Configuring your Boot Tracks 137
- Backing Up your Diskettes 139

Running Application Programs **141**

- Determining Diskette Format 141
- Making an Exact Copy of a Master Diskette 142
- Altering the Copy 142
- One Final Note on Running Application Programs 144

Section 2
p-System
Editor

147

Introduction

149

Overview	149
The Edit Menu	150
The Window Into the File	151
The Cursor	151
Editing Text	152
Notation Conventions	152

Lesson 1: Beginning to Use the Editor

154

Inserting Text	154
Moving the Cursor From the Edit Menu	155
Leaving the Editor	157

Lesson 2: More Advanced Instructions

160

I(nsert	160
D(elete	161
F(ind and R(eplace	162
Controlling the Margins	166

Reference Section

169

Accepting or Rejecting Commands	169
Copy Buffer	169
Entering and Exiting the Editor	170
Cursor Movement Keys	172
Cursor Movement Commands	172

Direction Indicator	173
Repeat Factors	173
Editor Options	173
The Editor Commands Listed by Function	175
The Editor Commands	176
A(djust	176
C(opy	177
D(elete	178
F(ind	179
I(nsert	181
J(ump	184
K(olumn	184
M(argin	185
P(age	187
Q(uit	187
R(eplace	190
S(et	192
V(erify	198
X(change	198
Z(ap	199

Section 3

PRINT

201

Introduction

203

Simple Uses of PRINT 203

PRINT Options

206

Controlling the Page Layout 206

Printer Commands Within the Text File 212

Escape sequences within your file 212
 Command Lines 212

Printing your File 219

I(nput file 219
 O(utput 220
 S(top and U(se Formfeed 220
 A(dvance the printer 221
 G(o 221
 Q(uit 222
 M(ake a script file 222
 Summary of Menu Items 225
 Summary of Command Lines 226
 Summary of Escape Sequences 227

PART 1
CP/M Plus

Introduction

Three CP/M Plus disks are supplied with the **Osborne Executive**. The **System Disk** contains the operating system's built-in commands and programs such as COPYSYS and PIP. The built-in commands are part of the system tracks on your program diskettes. Volume 1 of the *Osborne Executive Guides* describes how to use the built-in commands.

The CP/M Plus **General Utilities Disk** and the **Advanced Utilities Disk** contain transient utilities. These are commands stored as .COM files on diskette rather than on the system tracks. To use a transient command, the diskette containing it must be in the active disk drive. Chapter 1 in this volume describes the CP/M Plus transient utilities.

If necessary, review the material in *Volume 1 —Mastering the Osborne Executive*, before continuing from here. You'll need a basic acquaintance with CP/M Plus operations to best use the present material.

For convenience and quick reference, the following table lists CP/M Plus control characters. These commands function when the **A>** or **B>** prompt is on screen.

Control Characters

- ^H** backspaces and deletes one character position.
- ^A** advances cursor one character to the right.
- ^B** moves cursor to beginning of the command line without affecting the contents of the line. If the cursor is already at the beginning, **^B** moves it to the end of the line.
- ^F** inserts one character space at the cursor.
- ^G** moves cursor one character to the left.
- ^I** moves cursor to the next tab stop. Tab stops are automatically set at every eighth column. This control command has the same effect as the TAB key.
- ^R** displays a # at the cursor position, then moves the cursor to the next line and retypes the previous command line.
- ^K** deletes from cursor position to end of the line.
- ^U** deletes entire line the cursor is in and moves the cursor one line down.
- ^X** deletes entire line the cursor is in by backspacing over it. Returns cursor to beginning of current line.
- ^M** inserts a line feed and carriage return, then sends the current command line to be processed.
- ^E** inserts a line feed and carriage return on screen, but doesn't send the command line for processing.
- ^J** inserts a line feed and sends the current command line to be processed.
- ^R** redisplayes the current command sequence on the next line without deleted characters.

- ^P** sends data on screen to a printer through the RS232 connector. The bell rings when this command activates a printer connected to the computer. A second **^P** turns the printing OFF.
- ^S** temporarily halts screen display. Pressing **^S** again resumes the display.
- ^Q** resumes screen scrolling after a **^S**.
- ^Z** ends input from the system console (keyboard). The PIP and ED commands use **^** to indicate that the end of a file has been transmitted by the keyboard.
- ^C** terminates program execution and displays the **A>** or **B>** prompt if the cursor is at the beginning of the command line.

CP/M Plus Utilities

This chapter describes the programs on your CP/M Plus Utilities diskette. The chapter progresses from the simplest utilities to more specialized ones used in programming.

Before starting this chapter, you should be familiar with the CP/M Plus built-in commands. They're described in *Volume 1—Mastering the Osborne Executive*, of the *Osborne Executive Guides*.

When you're ready to use this chapter, try the examples on your computer as you come to them in the text. For more detailed information about any of the CP/M Plus utilities, see the *Osborne Executive Reference Guide*.

What are Utilities?

CP/M Plus utilities are tools for managing the computer. They cover a range of tasks from copying files to assembling and debugging programs. There are two kinds of utilities: **built-in** and **transient**.

Built-in utilities are available whenever CP/M Plus is in the computer's memory. These six utilities include **DIR**, **ERASE**, **RENAME** and the others you've used in Volume 2.

Transient utilities are available only when the CP/M Plus Utilities diskette is in the active disk drive (drive A).

In this chapter you'll learn how to use the transient utilities and commands. We've divided the chapter as follows:

- Section 1, *Startup Utilities*, describes those which are useful when you begin to use the computer or a new diskette.
- Section 2, *Diskette Attributes*, explains diskette attributes and how to examine information in files.
- Section 3, *File Handling With PIP*, explains how to copy, transfer, and rearrange files.
- Section 4, *Communicating With External Devices*, discusses data transfer between the Osborne Executive and other devices.
- Section 5, *ED, the CP/M Plus Line Editor*, describes the CP/M Plus editor for creating and changing text files.
- Section 6, *File Processing and Data Handling*, describes input/output switching between files and commands placed in files for later execution.
- Section 7, *Running Assembly Language Programs*, explains how to run assembly language programs.

Section 1: Startup Utilities

HELP
DATE

INITDIR (*Initialize Directory*)
SETDEF

This section explains how to:

- use the CP/M Plus HELP features
- set the system clock
- prepare diskettes to record date and time
- find and display data

Insert the CP/M Plus General Utilities Disk in drive A and press RETURN. The computer displays this screen:



The **A>** prompt means that the Executive has loaded CP/M Plus and is waiting for a command.

Type **[D][I][R]**.

Help Note:

If you make a typing error, use the LEFT ARROW key to backspace over it. Then type the correction.

Press **[RETURN]** to display a list of files on the CP/M Plus General Utilities Disk:

```
CP/N V3.0 Loader
Copyright (C) 1982, Digital Research

60K TPA

Osborne Executive CP/M 3.0
Alpha test version X3.838

A>dir
A: PIP      COM : DATE   COM : DEVICE  COM : DIR     COM : ERASE   COM
A: RENAME  COM : SET     COM : SETDEF  COM : SHOW   COM : SUBMIT  COM
A: TYPE    COM : HELP   HLP : HELP   COM : ED     COM
SYSTEM FILE(S) EXIST
A>█
```

The HELP Command

We're interested in the file named **HELP.COM** and the information it has about CP/M Plus.

Type **H E L P** and press **RETURN**. The computer responds with a list of topics:

```
HELP UTILITY V1.1

At "HELP>" enter topic {,subtopic}...

EXAMPLE: HELP> DIR EXAMPLES

Topics available:

COMMANDS  CNTRLCHARS  COPYSYS  DATE      DEVICE    DIR
DUMP      ED              ERASE    FILESPEC  GENCOM    GET
HELP      HEXCOM         INITDIR  LIB        LINK       MAC
PATCH    PIP (COPY)    PUT      RENAME    RMAC      SAVE
SET       SETDEF        SHOW     SID        SUBMIT    TYPE
USER      XREF

HELP> █
```

The items on this list are CP/M Plus functions or commands. The HELP utility we're using is a quick way to find out more about these features while using the computer.

The HELP prompt appears in the lower left corner of the screen, like this:

```
HELP>
```

This means the computer is waiting for you to choose one of the listed topics. We'll choose the topic **RENAME** and ask for more information about it.

Type **R****E****N****A****M****E**. It will appear on screen next to the **HELP** prompt:

```
HELP>RENAME
```

Press **RETURN** to display a brief summary of the **RENAME** utility. The information on screen will give you an idea of what **RENAME** is and how it works. That's the purpose of the **HELP** utility—a quick orientation when you need it!

You can ask for more detailed information about **RENAME**. Notice the line on the screen which says:

```
ENTER.subtopic FOR INFORMATION ON THE FOLLOWING SUBTOPICS:  
  
EXAMPLES
```

EXAMPLES is shown here as a **subtopic** of the **RENAME** utility. Let's see these examples of how to use **RENAME**.

Type **.EXAMPLES**. Be sure to include the period (.) at the beginning of the word.

Press **RETURN**.

This time the screen displays examples of **RENAME** as used in several command lines. When you've read through them, press **RETURN** to display a few more.

Finished with the RENAME examples? Press **RETURN** to display the CP/M Plus **A>** prompt.

Help

Command:

Step-by-Step

The CP/M Plus General Utilities Disk is in drive **A**:

1. Type **H E L P** after the **A>**.
2. Press **RETURN** to display a list of topics.
3. Type the topic you're interested in.
4. Press **RETURN** to display a summary of the topic.
5. For information on a subtopic, type a period and the subtopic name.
6. Press **RETURN** to display the information.
7. To exit to the **A>**, press **RETURN**.

As you become familiar with topic names, you can skip Step 2 by typing a space and the topic name after **HELP**.

You can also change the list of topics shown by the **HELP** command. The *Osborne Executive Reference Guide* has instructions for doing this.

The DATE Command

The **DATE** utility will set time and date when you turn the Osborne Executive ON or reset it. You can then display time and date on screen or record it on your diskettes and printouts.

Set the Clock

The CP/M Plus General Utilities Disk must be in drive **A**.
With the **A>** on screen, type:

A> **D A T E** **SPACE BAR** **S E T**

Press **RETURN** to display this message:

Enter today's date: (MM/DD/YY)

Enter the month, day, and year in double-digit format (for example, 05/23/84). Include a slash (/) between them. When you're ready, press **RETURN**. This message appears at the bottom of your screen:

Enter the time: (HH:MM:SS)

Enter the time as hours, minutes, and seconds in double-digit format (for example, 11:23:30). Include a colon (:) between each entry. When you've completed the entries, press **RETURN** to display a new message:

Press any key to set time

Press any key to log the date and time into the computer. This sequence of steps takes a few moments. If you need to set the clock **exactly** to the second, take this into account when you enter the time.

There's also a faster way to set the time. Type **D A T E** when the **A>** is on screen. Then enter the date and time in double-digit format. Here's an example:

A> **DATE 06/15/83 15:25:13**

After typing the date and time, press **RETURN**. Then press any key to log it in.

DATE

Command: Step-by-Step

The CP/M Plus General Utilities Disk is in drive A:

1. Type `D A T E` `SPACE BAR` `S E T` after the `A>`.
2. Press `RETURN`.
3. Enter month/day/year in double-digit format (for example, 02/23/84). Include a slash between entries.
4. Press `RETURN`.
5. Enter hour:minute:second in double-digit format (for example, 11:23:30). Include a colon between entries.
6. Press `RETURN`.
7. Press any key to set time.

Time Display on Screen

Setting the time when you begin using the computer lets you check it afterward. To **display the current date and time**:

1. Type `D A T E` after the `A>`.
2. Press `RETURN`.

The screen will show the date and time, as in this example:

```
Tue 04/25/83 14:58:34
```

To **display the clock itself on the command line**:

1. Type `D A T E` `SPACE BAR` `C` after the `A>`.
2. Press `RETURN`.

This sequence will show the clock running. To terminate the display, press any key.

Time Stamping Files

Once you've set the Executive's clock, you can record the time and date on your data diskettes. This is called **time stamping**. It's convenient for keeping track of when data is entered or changed in files.

INITDIR: Preparing Space for the Time Stamp

The first step in time stamping a diskette is to prepare space for the date and time. **INITDIR** (Initialize Directory) is the utility which does this. You need to prepare a diskette this way only once, preferably right after formatting it.

The CP/M Plus General Utilities Disk must be in drive **A**. Insert a diskette containing data files into drive **B**. Then type:

```
A> I N I T D I R [SPACE BAR] B :
```

Press **RETURN** to display this message:

INITDIR WILL ACTIVATE TIME STAMPS FOR SPECIFIED DRIVE.
Do you want to re-format the directory on drive: B (Y/N)?

Press **Y** (for Yes), then press **RETURN**. The **A>** prompt will appear after a few moments. This means space has been set aside on the diskette for the time and date.

In this example we've prepared space for time stamping on a blank diskette. **INITDIR** will also prepare space on a diskette containing files, if there's room available in the file directory.

SET: Recording Date and Time

You've logged the date and time into the computer and prepared space on the diskette for the time stamp. To actually record time and date on the diskette in drive **B**, we'll use the **SET** utility. The file directory will then include a time and date record.

There are three time stamping options: **ACCESS**, **CREATE**, and **UPDATE**.

The **ACCESS** option will record when a file was last read. Each time a file is read, the time stamp changes accordingly.

The **CREATE** option will record only when a file was created.

The **UPDATE** option will record when a file was last changed or updated.

ACCESS and **CREATE** can't be used together, but you can use **UPDATE** with either of them.

To time stamp the diskette in drive **B**, the **A>** must be on screen. In this example, we'll choose the **CREATE** time stamping option. Type:

```
A> S E T SPACE BAR B : SPACE BAR C R E A T E  
= O N I
```

Press **RETURN**. You'll see a **label** or summary about the diskette in drive **B**. Like this:

Label for drive B:				
Directory <u>Label</u>	Passwrds <u>Reqd</u>	Stamp <u>Create</u>	Stamp <u>Access</u>	Stamp <u>Update</u>
B:Label	off	on	off	off

Column three of this screen shows that you've turned ON the CREATE option. Let's add the UPDATE option, too.

Type:

```
A> S E T SPACE BAR B : SPACE BAR U P D A T E
    = O N I
```

Press **RETURN**. The drive **B** label or summary appears again:

Label for drive B:				
Directory <u>Label</u>	Passwrds <u>Reqd</u>	Stamp <u>Create</u>	Stamp <u>Access</u>	Stamp <u>Update</u>
B:Label	off	on	off	on

The last column shows that the UPDATE option is also ON. The diskette in drive **B** is now time stamped. Its directory will show when its files are created and when they're changed or updated.

How about changing or turning OFF a time stamp option? It's the reverse of what you've just done. For example, to turn OFF the CREATE option, type:

```
A> S E T SPACE BAR B : SPACE BAR C R E A T E
    = O F F I
```

and press **RETURN**. The drive **B** label which appears on screen will show "Stamp Create Off." Now you can specify the ACCESS time stamp or none at all.

Time Stamping: Step-by-Step

The CP/M Plus General Utilities Disk is in drive A. A formatted data diskette is in drive B.

1. Use DATE to set the computer's clock.
2. Use INITDIR to prepare space on the data diskette for time and date.
3. Use SET to specify options and implement time stamping.

Displaying Time and Date

After you've implemented time stamping on a diskette, the file directory will show the time record if you ask for it. To include the time record in the directory screen, type:

```
A> DIR [SPACE BAR] B : [SPACE BAR] [DATE]
```

and press **RETURN**. You'll see a directory similar to this example:

```
Directory for drive B: User 0
```

Name	Bytes	Recs	Attributes	Prot	Update	Access
JONES	TXT 2k	1 DIR	RW		NONE	
WORK	COM 4k	2 DIR	RW		NONE	
Total Bytes =		20K	Total Records =	0	Files Found =	11
Total 1K Blocks =		0	Used/Max Dir Entries For Drive B:			4/64

This directory has information about a diskette with two files: JONES.TXT and WORK.COM. The last two columns show whether time stamping is in effect.

After time stamping a diskette, **be sure to set the computer's clock** with the DATE command each time you use the diskette. This will keep the time stamp correct and up to date.

SETDEF: Finding and Displaying Files

The **SETDEF** utility specifies how the computer searches for and displays files. We'll describe frequently-used SETDEF features in this section. The *Osborne Executive Reference Guide* has a complete discussion of SETDEF.

With the CP/M Plus General Utilities Disk in drive A, type:

```
A> S E T D E F RETURN
```

The Executive responds with this screen:

```
Drive Search Path:
1st Drive          - Default
Search Order       - COM
Temporary Drive    - Default
Console Page Mode  - On
Program Name Display - Off
```

These are CP/M Plus **default settings**. It means the computer searches for a file in this sequence **unless you change it**.

Here's what these entries mean:

1st Drive

the computer searches first for programs on the default drive, drive A.

Search Order	the computer will search for a .COM type file unless you instruct otherwise.
Temporary Drive	files are stored here temporarily during processing.
Console Page Mode	when ON, displays information in 24-line segments (one screenful). Press any key to continue the display.
Program Name Display	when ON, displays name and location of specified program.

You can change these default settings with **SETDEF** commands. For example, to place temporary files on the diskette in drive **A** instead of on the work diskette in drive **B**, type:

```
A> S E T D E F SPACE BAR [ ] T E M P O R A R Y
    [ ] [ A ] [ ] [ ]
```

and press **[RETURN]**.

SETDEF can also specify file type to be searched for when you execute a program. Typing the command, **TYPE JONES**, for example, instructs the Executive to search for a command program named JONES.COM. The search order can be changed so that the system searches for different types of files in sequence. The following example instructs the Executive to look for a COM file and, if it doesn't find one, to search for a SUB file:

```
A> SETDEF [ORDER=(COM,SUB)]
```

Finally, SETDEF will turn the system page mode OFF or ON for the CP/M Plus utilities. Page mode ON means there will

be a pause after each full screen of information. In this mode, press any key to continue to the next screen.

Page mode OFF means the screen will scroll continuously until the end of the display.

To turn page mode OFF, type:

S E T D E F [SPACE BAR] **[] N O** [SPACE BAR] **P A G E** **[]**
and press **[RETURN]**.

To turn page mode ON, type:

S E T D E F [SPACE BAR] **[] P A G E** **[]**
and press **[RETURN]**.

SETDEF Command Summary

Press RETURN after typing any of these commands:

SETDEF [TEMPORARY=A]	places temporary files on disk in drive A during processing.
SETDEF [ORDER=(filetype)]	instructs the Executive to search for file type name.
SETDEF [ORDER=(filetype1,filetype2)]	instructs system to search for file type 1, then file type 2.
SETDEF [NO PAGE]	causes continuous screen scrolling.
SETDEF [PAGE]	displays information by single screens. Press any key to continue display.

Section 2: Diskette Attributes

*SET Attributes
Passwords*

*SHOW
Directory Options*

With CP/M Plus, you can assign various **attributes** to your diskettes. Attributes are optional features which identify a diskette and specify how it can be used or who can use it. Diskette attributes include name, read/write status, passwords, user numbers, and user areas.

This section explains how to:

- set diskette attributes
- display this information on screen

Insert the CP/M Plus **General Utilities Disk** in drive **A** and a formatted work diskette in drive **B**. Press `[RETURN]` to display the **A>** prompt.

Naming a Diskette

SET is the general command for assigning diskette attributes. We'll use it first to **name** the diskette in drive **B**.

The name you assign to a diskette will appear in the diskette's directory. It's a convenient way to keep track of what the diskette contains. The name you assign to a diskette can be up to eight characters long, but can't have spaces between the characters.

As an example, we'll assign the name **TEST1** to the diskette in drive **B**. Type:

```
A> S E T SPACE BAR B : SPACE BAR
    [ N A M E = T E S T 1 ]
```

Press **RETURN** to display this screen:

Label for drive B:				
Directory Label	Passwds Reqd	Stamp Create	Stamp Access	Stamp Update
-----	-----	-----	-----	-----
B:TEST1	off	off	off	on
A>				

This screen shows the **label**, or list of attributes, for the diskette in drive **B**. The first column of the screen shows that the diskette in drive **B** is named **TEST1**.

Making a Diskette Read Only

Information can normally be written onto and read from a diskette. This characteristic is called **Read/Write**. You may wish to make a diskette **Read Only**. This means the computer can **read data** from the diskette but not write onto it. You can't create, delete, or change any files on a Read Only diskette. Making a diskette Read Only protects the data on it from being changed.

One way to make a diskette Read Only is to place a write-protect tab on it. Another way to do it is with the **SET** command.

To make the diskette in drive **B** Read Only, type:

```
A> S E T SPACE BAR B : SPACE BAR [ ] R [ ] O [ ]
```

Press **RETURN**. The computer responds with this message:

```
Drive B: set to Read Only (RO)
```

Now you'll be able to read the files on the diskette, but not enter or change any data.

How do you change the diskette back to Read/Write? Type this **SET** command:

```
A> S E T SPACE BAR B : SPACE BAR [ ] R [ ] W [ ]
```

Press **RETURN** to display the message:

```
Drive B: set to Read Write (RW)
```

The files are fully available again for entering or changing data.

Passwords

Passwords are valuable for preventing unauthorized access to data. With the **SET** utility, you can assign passwords to individual files or an entire diskette. Passwords can be up to eight characters long, but there can't be spaces between the characters.

Note: Keep a record of your passwords and the files or diskettes they're assigned to! If you forget a password, you may lose access to your data.

Diskette Passwords

A password assigned to a diskette restricts

- access to the diskette
- use of the SET command for that diskette

A diskette must have a name **before** it's assigned a password. As an example, we'll assign the password **SAFE** to the diskette you named TEST1.

Type:

```
A> [S][E][T][SPACE BAR][B]:[SPACE BAR]
    [I][P][A][S][S][W][O][R][D]=[S][A][F][E]
```

Press **[RETURN]** to display this screen:

```
Label for drive B:

Directory      Passwds   Stamp    Stamp    Stamp
Label          Reqd     Create   Access   Update
-----
B:TEST1      .      off      off      off      on

Password = SAFE
A>
```

The password **SAFE** is now assigned to the diskette named TEST1. The Executive won't carry out any **SET** command for this diskette unless the password is entered first.

If you issue a SET command for a password-protected diskette, you'll see this message:

```
Directory Label
Password?
```

Type the password `S A F E` and press `RETURN`. This will give you access to the TEST1 diskette.

To **remove** the password from TEST1, type:

```
A> S E T SPACE BAR B : SPACE BAR
    I P A S S W O R D =
```

then press `RETURN`.

File Passwords

Passwords can be assigned to **individual files** as well as an entire diskette. There are two steps for doing it.

1st Step. Prepare the diskette with this command (for drive B):

```
A> S E T SPACE BAR B : SPACE BAR
    I P R O T E C T = O N ]
```

Press `RETURN` to display the label screen for drive B:

```
Label for drive B:
```

Directory Label	Passwds Reqd	Stamp Create	Stamp Access	Stamp Update
-----	-----	-----	-----	-----
B:TEST1	on	off	off	on

```
A>
```

2nd Step. Assign the file password. For example, to assign the password ONLYN to a file named ACCOUNT, you would type:

```
A> S E T SPACE BAR B : A C C O U N T SPACE BAR
    P A S S W O R D = O N L Y N ]
```

and press **RETURN**. The computer responds with

```
B:ACCOUNT Protection = READ, Password = ONLYN
```

The ACCOUNT file now can't be read or changed unless the password ONLYN is entered first.

You can use the characters * and ? to assign a password to *all* files of a certain type. For example, to assign the password ONLYN to all COM files in drive B, type:

```
A> S E T SPACE BAR B : SPACE BAR * . C O M
    SPACE BAR P A S S W O R D = O N L Y N ]
```

then press **RETURN**. This would prevent access to any COM file unless the password ONLYN was entered first.

To **remove** a password from a file in drive B, type:

```
A> S E T SPACE BAR B : SPACE BAR
    filename.typ P A S S W O R D = O F F ]
```

then press **RETURN**.

Password Options

It's sometimes convenient to restrict some file activities but not others. You might wish, for example, to protect a file from

deletion but allow it to be read and changed. You can use passwords to protect a file **selectively** this way:

Function	Description
READ	Password required for reading, copying, writing, deleting, or renaming the file.
WRITE	Password required for writing, deleting, or renaming the file.
DELETE	Password required to delete or rename the file.

The following example assigns the password ONLYN to a file named ACCOUNT. The file can be read freely, but the password ONLYN is required to change, delete, or rename it.

```
A>SET ACCOUNT [PASSWORD=ONLYN,
PROTECT=WRITE]
```

Access to System Files

Files are associated with particular user numbers and areas on diskette. Normally, a user number has access only to files assigned to it. The **SET** command can designate files as **System** files. System files can be executed by any user number.

The CP/M Plus **General Utilities Disk** in drive **A** includes a file called **RENAME.COM**. We'll use **SET** to make it a System file. Type:

```
A> S E T SPACE BAR R E N A M E . C O M SPACE BAR
   S Y S I
```

and press **RETURN**.

RENAME.COM is now a System file, executable from any user number. This is only an attribute, however. RENAME's file type is still COM.

The DIRS command will list RENAME in the directory of System files from user number 0.

SET Command Summary

These SET commands assign attributes for the diskette in drive B.

SET B: [NAME=TEST1] names the diskette TEST1

SET B: [RO] makes the diskette Read Only

SET B: [RW] restores diskette to Read/Write status

SET B: [PASSWORD=SAFE] assigns the password SAFE to the diskette

SET B: ACCOUNT [PASSWORD=ONLYN] assigns the password ONLYN to the file named ACCOUNT

SET B: [PASSWORD=] cancels password from diskette

SET B: filename.typ[PASSWORD= OFF] cancels password from file

SHOW and DIR: Checking Diskette Contents

SET commands establish diskette attributes. SHOW and DIR display what they are. SHOW displays the read/write status of a diskette and the space available on it.

The CP/M Plus Systems Disk should be in drive **A** and a work diskette in drive **B**. Type:

A> **S H O W**

Press **RETURN** to display a message like this one:

```
A: RW, Space:    11k
```

This screen shows the diskette in drive **A** is read/write and has 11k of available space. **Read/write** means you can read data from and write it onto the diskette. The available space (11k) is equivalent to about 5 single-spaced typewritten pages.

To check the status of the diskette in drive **B**, type:

A> **S H O W SPACE BAR B :**

then press **RETURN**.

Displaying a Diskette's Name

SHOW will also display a diskette's name and additional information about it. Type:

A> **S H O W SPACE BAR B : I L L A B E L I**

and press **RETURN**. You'll see a screen similar to this one:

```
Label for Drive A
```

Directory	Passwds	Stamp	Stamp		
Label	Reqd	Create	Update	Label Created	Label Updated
----	----	-----	-----	-----	-----
TEST1	on	on	on	05/12/83 15:43	06/14/82 09:30

Here's what these entries mean:

- Directory Label** shows the diskette's name. The diskette in this example is named TEST1.
- Passwds Req'd** ON means that a password applies to the diskette or files on it. OFF means the diskette has no passwords.
- Stamp Create** ON means the diskette has been time stamped. OFF means there's no time stamping for this diskette.
- Stamp Update** ON indicates that time stamping is active, OFF that it's not.
- Label Created** shows when the diskette name was created.
- Label Updated** shows when the diskette name was last changed.

Displaying User Numbers

SHOW will display user numbers and files associated with them. Type:

A> **S H O W** **S P A C E B A R** **[U S E R]**

Press **RETURN** for a display like this example:

```
Active User:      3
Active Files:    0 7 11
A:# of files:    9 3 8

A: Number of free directory entries: 29
```

These entries apply to drive A:

Active User	your user number.
Active Files	user numbers assigned for the diskette.
A:# of files	the number of files associated with each user number.
A:Number of free directory entries	the number of directory entry spaces still available.

Detailed File Directories

The **DIR** and **DIRS** commands have options for detailed directories.

For the most complete information about the size and number of files on a diskette, type:

A> `DIR SPACE BAR B . T F U L L T`

Press `RETURN`. You'll see a directory like this example:

Directory for Drive A: User 3

Name	Bytes	Recs	Attributes	Prot	Update	Access
JONES TXT	2k	1	Sys RW	None		
EXPER COM	6k	48	Dir RW	Read		04/14/83

Total Bytes = 8K Total Records = 49 Files Found = 2
 Total 1k Blocks = 4 Used/Max Dir Entries for Drive A: 2/26

These are the full directory entries:

Name	file names are listed in this column. The example shows two files: JONES TXT and EXPER COM
Bytes	kilobytes of disk space used by each file
Recs	number of 128-byte records used by each file
Attributes	attributes established for each file
Prot	indicates whether a file is password protected
Update	shows when file was last changed
Access	shows when file was last accessed

The bottom of this directory summarizes space used and the number of files and records on the diskette.

Directory Options

DIR can be used with the directory options shown below. To use these options, type the command indicated and press RETURN.

<u>DIR Command</u>	<u>Displays</u>
A>DIR [DATE]	File names and time/date stamps
A>DIR [DRIVE=ALL]	Directories for all drives
A>DIR [DRIVE=(A,C)]	File names on drives specified. In this example, drives A and C.

A>DIR [USER=ALL] All file names on the diskette, including those of other users.

A>DIR [USER=5] Names of files assigned to a specific user. In this example, user 5.

**A>DIR [EXCLUDE]
*.COM** All files names except those specified. This example excludes all COM files from the directory.

You can also combine these options in DIR commands. Here's an example:

A>DIR B: [USER=7 FULL EXCLUDE] *.TXT

This command displays the directory of files belonging to user 7 in drive **B** except TXT files.

Section 3: File Handling With PIP

<i>PIP</i>	<i>PIP Options</i>
------------	--------------------

PIP, the Peripheral Interchange Program, is a utility for copying files from place to place on a diskette or from one diskette to another. PIP copies both the file and its attributes.

The PIP command has this general format:

A>PIP destination:filename=source:filename

These are the PIP command entries:

destination: the drive into which you're copying. If you're copying from place to place on the same diskette, the destination and source will be the same.

filename the name you're using for the copy. The copy can have the same name as the original file or a different one if you wish.

source: the drive from which you're copying.

filename the file you're copying.

Insert the CP/M Plus **General System Disk** in drive **A** and a work diskette in drive **B**.

Copying Individual Files

As the first example, we'll copy the HELP.COM file from the utilities diskette to the work diskette. Type:

```
A> P I P SPACE BAR B : H E L P . C O M =  
A : H E L P . C O M
```

Press **RETURN** to activate PIP. When the copy's made, the **A>** prompt will appear on screen.

The drive **B** directory will list **HELP.COM** after it's been copied. To check it, type:

```
A> D I R SPACE BAR B :
```

and press **RETURN**.

Copying Several Files

PIP will also copy a sequence of files.

First, type PIP. Like this:

```
A> P I P
```

Press **RETURN**. The asterisk (*) which appears on screen is the PIP prompt. It means PIP is ready for instructions. The display looks like this:

```
A>PIP  
CP/M3 PIP VERSION 3.0  
*
```

Here's an example of the PIP sequence using three imaginary files: **JBACCT**, **ACCOUNT6**, and **JONES.TXT**. It begins at the PIP prompt (*).

Step 1. To copy **JBACCT** from drive **A** to drive **B** and rename it **LLACCT**, type:

***B:LLACCT=A:JBACCT** and press RETURN.

Step 2. To copy **ACCOUNT6** from drive **A** to drive **B** and rename it **XYACCT**, type:

***B:XYACCT=A:ACCOUNT6** and press RETURN.

Step 3. To copy **JONES.TXT** from drive **B** to drive **A**, type:

***A:JONES.TXT=B:JONES.TXT** and press RETURN.

Step 4. To return from the PIP prompt (*) to the **A>** prompt, press RETURN.

The PIP sequence would look like this on screen:

```
A>PIP
*B:LLACCT=A:JBACCT
*B:XYACCT=A:ACCOUNT6
*A:JONES.TXT=B:JONES.TXT
*
A>
```

Copying Groups of Files

PIP is very convenient for copying groups of related files using the inclusive CP/M characters ? and *. These are the "wildcard" characters described in *Volume 1—Mastering the Osborne Executive*.

For example, to copy all files on drive **A** whose names begin with **AUTOST**, type:

```
A> PIP [SPACE BAR] B := A : AUTOST *
```

Press **RETURN**. The **A>** prompt appears on screen when copying is completed.

To copy **all** files from drive **A** to drive **B**, type:

```
A> PIP [SPACE BAR] B := A : * . *
```

and press **RETURN**.

Using wildcard characters in a PIP command is efficient for copying data files onto a diskette. However, it's of limited value for copying program diskettes because it doesn't copy the CP/M Plus operating system itself.

Merging Files

PIP copies files separately and in groups. It will also **copy and merge** different files into a single one. (This is called **file concatenation**). The destination file can be on the same diskette as the source files or a different one.

Before merging separate files into a larger one, use **DIR** and **SHOW** to check that there's space available on the destination diskette.

Here's an example of file merger using PIP. The source files in drive **B** are named **TEXT1**, **TEXT2**, and **TEXT3**. The destination file in drive **A** is named **CHAPTER1**.

```
A> PIP A:CHAPTER1.TXT =
      B:TEXT1.TXT,TEXT2.TXT,TEXT3.TXT
```

To execute this command, you would press **RETURN**. The **A>** would appear when copying's completed.

PIP Options

Like other CP/M Plus utilities, PIP has various options. Options are indicated by **letters** in PIP commands.

The format for PIP options is to enclose them in square brackets, [], as the last item in the source and destination entries. If there's more than one option in the brackets, include a space between each one.

This example shows a PIP command with options:

```
A>PIP A:[G5]=B:ACCOUNT*[A G2 E]
```

In this example,

A is the destination drive.

[G5] is an option. It specifies that the copies are assigned to user 5.

B is the source drive.

ACCOUNT* specifies that all files beginning with ACCOUNT should be copied.

[A G2 E] are three options which identify the files to be copied. The options are described in the following list of PIP options.

List of PIP Options

A (Archive)	copies only files changed since last being copied. This option applies only to the source diskette and only when time stamping is in effect.
--------------------	--

- Example: to copy all files in drive **A** which have been changed since last being copied, type:
- A>PIP B:=A:*. *[A]**
and press RETURN.
- C (Confirm)** displays file name and asks you to confirm it before making the copy. Applies only to the source diskette. This option is useful for copying only certain files of a particular type.
- Example: to display the name of each TEXT file for confirmation before it is copied, type:
- A>PIP B:A:*.TXT[C]**
and press RETURN.
- E (Echo)** displays contents of the source file as it's being copied. Applies only to text (ASCII) files on the source diskette.
- Gn**
(Get or Go To
User Number) applies to either source or destination diskette. On the source diskette, specifies user area *n* to copy *from*. On the destination diskette, specifies user area to copy *to*.
- Example: to copy the SMITH file from user area 7 on drive **A** to user area 3 on drive **B**, type:
- A>PIP B:SMITH [G3]=A:SMITH [G7]**
- R**
(Read SYS Files) specifies copying of system files and their attributes. Applies only to source diskette.

V (Verify)	checks that copying is successful. Displays error message if not. Applies to either source or destination diskette.
W (Write over)	copies a file onto a Read Only (RO) destination file. Applies only to source files. If copying several source files onto Read Only destinations, include a W after each source file.
Z (Zero)	strips a file of its parity bits. Applies only to source files. Useful for converting WordStar document files into non-document files by removing the high bit.

These are only a few PIP options. The *Osborne Executive Reference Guide* lists many others.

Copying To and From External Devices

The PIP commands described so far copy files from place to place on a diskette or between diskettes. PIP will also copy data between the Osborne Executive and external devices such as a printer.

To copy data this way, the external device must be designated in the PIP command. Here are several examples of external device designations. The *Osborne Executive Reference Guide* includes a complete list.

CON: (Console)	designates the keyboard for input and the video screen for output
AUX: (Auxiliary)	designates a modem, printer, another computer, or any other device
LST: (LIST)	designates an output device, usually a printer.

Command for copying data to and from external devices follow the general PIP format. This command, for example, sends data from the keyboard to the printer:

A>PIP PRN:=CON:

This PIP command sends to the printer anything typed at the keyboard. **CTRL-Z** terminates the command.

The following example copies from an auxiliary device such as a computer to the JONES.TXT file in drive **B**:

A>PIP B:JONES.TXT=AUX:

CTRL-Z terminates this command. PIP has many external device options. Several examples are shown below, and the *Osborne Executive Reference Guide* has a complete list.

PIP External Device Options

Dn
(Delete past
column n)

specifies that the destination device copy only up to column n. Applies only to source entries in the PIP command.

Example: To print the first 40 columns of each line in the file ORDERS.TXT, type:

A>PIP LST:=B:ORDERS.TXT[D40]
and press RETURN.

F
(Remove Form
Feeds)

removes all form feeds included in the source file. Applies only to the source file.

Pn specifies number of lines **n** per page.
(Page length, Applies only to the source file. Use **Pn**
number of files) alone if the source file has no form
feeds. Otherwise, use it and **F** together.

Example: To print the file PROP.TXT
with 40 lines per page, type:

A>LST:=B:PROP.TXT[F P40]
and press RETURN.

PIP Summary

PIP is a highly versatile utility for handling files. Here are some guidelines for using it efficiently.

- The general PIP command format is
A>PIP destination:filename = source:filename
- The destination diskette must have enough room for the file(s) you're copying. Use DIR [SIZE] to check the size of the source file and SHOW to check space available on the destination diskette.
- When copying a file from place to place on the same diskette, rename the file or give it a new file type.
- PIP will copy a source file onto a destination diskette containing a file with the same name. PIP erases the file on the destination diskette after copying the source file.
- Before overwriting a "Read Only" file, PIP asks for confirmation. Press Y to overwrite or N to prevent it.
- PIP copies file attributes (SYS, DIR, RW, and RO), user number, and password with the file.

Section 4: Communicating With External Devices

Logical and Physical Devices
DEVICE Command

Section 3 described how **PIP** can copy data between the Osborne Executive and external devices. When data is transferred this way, it must be organized so the devices can handle it. The rules for organizing data transfer are called communication **protocols**.

Generally, the **SETUP** program establishes protocols between the Osborne Executive and such devices as printers or modems. *Volume 1—Mastering the Osborne Executive*, describes **SETUP**.

The **DEVICE** utility is available if **SETUP** can't establish the appropriate protocol. This section describes how to use **DEVICE** to:

- identify the characteristics of external devices
- set up the computer for data transfer

Physical and Logical Devices

Internal components like disk drives and the video monitor as well as external equipment like printers or modems are called **physical devices**. A physical device is an actual object which transmits data to and from a computer.

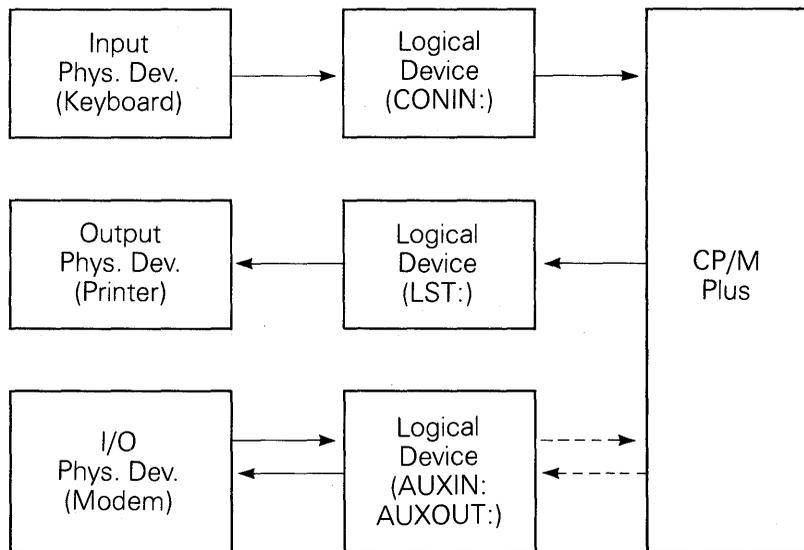
Computers classify physical devices into **logical devices** according to how they handle data. To communicate with a physical device, a computer *assigns it* to a logical device category.

CP/M Plus recognizes five logical devices:

CONIN and **CONOUT** refer to “console” devices such as the keyboard and video screen or a two-way teleprinter.

AUXIN and **AUXOUT** refer to “auxiliary” devices or anything other than the video screen and keyboard.

LST refers to the “list device,” usually a printer. The relationship between physical devices, logical devices, and CP/M Plus looks like this:



Data transfer between a computer and external devices can seem a bit intricate. The **DEVICE** command structures the exchange so the devices involved understand each other.

Displaying Device Assignments

The CP/M Plus Advanced Utilities Disk should be in drive A.
Type:

```
A>DEVICE
```

and press **RETURN**. You'll see a screen similar to the one below. The actual entries on your screen may differ depending on the devices connected to your computer.

```
A>DEVICE

Physical Devices:
I=Input,O=Output,S=Serial,X=Xon-Xoff
CRT  NONE IO      CEN  NONE IO      MODEM NONE IO
PRNTR NONE IO      IEEE NONE IO

Current Assignments:
CONIN: = CRT
CONOUT: = CRT
AUXIN: = Null Device
AUXOUT: = Null Device
LST: = CEN

Enter new assignment or press RETURN
```

The top line of this screen lists physical devices:

- I** indicates Input device.
- O** indicates Output device. The letters **IO** after a device name designate input/output.
- S** designates Serial data transfer between the computer and the device or between a logical device and a physical device.
- X** designates XON/XOFF communication protocol. This means the device sends a message to the computer whenever it's ready to receive data.

The second and third lines on screen are the logical device assignments. There are these entries:

CRT	indicating a terminal, usually some combination of keyboard and screen.
CEN	indicates a Centronics, or parallel, type printer.
MODEM	is any conventional MODulator/ DEModulator.
PRNTR	is the assignment of a serial type of printer.
IEEE	indicates the IEEE-488 IO port, or its equivalent.

The five NONE IO's indicate that there have been no actual device commitments yet.

Below the first two lines, the screen shows Current Assignments:

```
Current Assignments:  
CONIN: = CRT  
CONOUT: = CRT  
AUXIN: = Null Device  
AUXOUT: = CRT1  
LST: = LPT
```

Here's what these entries mean:

CONIN: = CRT	an external device named CRT, usually the keyboard and screen, is used for both input and output.
CONOUT: = CRT	

AUXIN: = Null Device	no physical device is assigned to the AUXIN logical device.
AUXOUT: = Null Device	a peripheral device used only for output.
LST: = CEN	a physical device (line printer) assigned to the LST logical device.

The last line on the DEVICE screen has the message:

Enter new assignment or press RETURN

Pressing RETURN confirms the entries on screen and displays the **A>** prompt. However, we're interested in changing device assignments and assigning new ones.

Assigning Physical Devices

Operating manuals for such devices as disk drives and printers specify their logical device assignment. They'll also have information about baud rates and communication protocols used by the equipment. You'll need this information to actually make logical assignments for your devices.

The Device Assignment screen should still be displayed. If it isn't, type **D E V I C E** after the **A>** and press **RETURN**.

We'll use DEVICE to change the **AUXIN:** device from NONE to MODEM. Type:

D E V I C E **SPACE BAR** **A U X I N : M O D E M**

and press **RETURN**. It looks like this on screen:

```
A>DEVICE AUXIN:=MODEM

Physical Devices:
I=Input,O=Output,S=Serial,X=Xon-Xoff
CRT  NONE IO    CEN  NONE IO    MODEM NONE IO
PRNTR NONE IO    IEEE NONE IO

Current Assignments:
CONIN: = CRT
CONOUT: = CRT
AUXIN: = Null Device
AUXOUT: = Null Device
LST: = CEN
```

This example shows the general format for assigning physical devices to logical devices. Include the logical device before the equal sign (=) and the physical device after it.

The following example assigns console output to both the CRT screen and a line printer:

```
Enter new assignment or press RETURN
DEVICE CONOUT:=CRT,LPT
```

These examples reassign external devices. The devices and their operating characteristics are already known to CP/M Plus.

DEVICE will also assign **new** physical devices and record their operating features.

1st Step. Identify the new device and indicate its operating characteristics.

For example, to identify a Xerox printer with a baud rate of 300 and designate it as an output device, type **DEVICE XEROX [300,O]** and press RETURN. On screen it looks like this:

```
Enter new device or press RETURN
DEVICE PRINTER [300,O]
```

2nd Step. Assign the physical device to a logical device. We'll assign it to **LST**, the typical logical device for a printer. Type:

DEVICE LST=PRINTER

We could also change the device's operating settings at this point. For example, to change the printer's baud rate from 300 to 110, type

DEVICE LST:=PRINTER [110,O]

After completing device assignments, press RETURN to confirm them and return to the **A>** prompt.

DEVICE commands must include logical devices. They can be designated as follows:

CON:	for CONIN: or CONOUT:
CONSOLE:	for CONIN: or CONOUT:
KEYBOARD:	for CONIN:
AUX:	for AUXIN: or AUXOUT:
AUXILIARY:	for AUXIN: or AUXOUT:
PRINTER:	for LST:

Displaying Device Information

To display a list of physical devices and logical device assignments type **DEVICE** and press **RETURN**.

There are also more limited **DEVICE** displays as follows. Press **RETURN** after each of these **DEVICE** commands for the corresponding display.

A>DEVICE NAMES	lists physical devices and their operating settings.
A>DEVICE VALUES	lists logical assignments of physical devices.
A>DEVICE CRT	lists operating characteristics of the actual CRT (screen and keyboard).
A>DEVICE CON	lists physical device assigned to CONIN and CONOUT .
A>DEVICE CONSOLE [PAGE]	lists console screen size by columns and lines.
A>DEVICE CONSOLE [COLUMNS=nn LINES=nn]	specifies screen size by number of columns (nn) and rows (nn).

Baud Rates and XON/XOFF Protocol

The rate of data transmission between the Osborne Executive and external devices varies with the device. Operating manuals for specific devices specify their baud rates.

CP/M Plus can operate with these baud rates:

50	75	110	131.5	150
300	600	1200	1800	2400
3600	4800	7200	9600	19200

This versatility means you can choose the most efficient transmission rate for each external device.

The XON/XOFF protocol uses two special ASCII characters: **XON** and **XOFF**. **XON** means "transmission ON"; **XOFF** means "transmission OFF." An external device sends XON to the computer when it's ready to receive data and XOFF when it's not. With XON/XOFF protocol in effect, the Osborne Executive checks for these signals before transmitting data to the external device.

NOXON signifies "No Protocol." In this case, the computer will transmit data without checking whether the peripheral device is ready to receive it.

Section 5: ED, The CP/M Plus Line Editor

ED is the CP/M Plus line editor. It's used to create and edit text line by line in ASCII program files. These program files contain letter and numeral text rather than machine language code.

Creating a File

Insert the CP/M Plus Advanced Utilities Disk in drive **A** and a formatted work diskette in drive **B**.

We'll use **ED** to create a file called **SAMPLE.ED** in drive **B**. With the **A>** on screen, type

```
A> [E][D][SPACE BAR][B]:[S][A][M][P][L][E].[E][D]
```

Press **[RETURN]** to open the file and display this screen:

```
NEW FILE
: *
```

The colon (:) and asterisk (*) are the ED prompt. The next step is to place the system in **Insert Mode**, so you can enter text. With the ED prompt on screen, press **[I]**, then press **[RETURN]**. This displays the line number 1.

Enter the lines shown below. Press **[RETURN]** after each line:

- 1: *THIS EXAMPLE SHOWS HOW TO
- 2: *ENTER TEXT
- 3: *USING THE ED UTILITY

After entering this text, enter ^Z to return to ED's command prompt, :* .

To save the SAMPLE.ED file, press ^E . This stores SAMPLE.ED in drive **B** and then displays the **A>** prompt.

SAMPLE.ED Summary

Here's a summary of file opening and text entry for SAMPLE.ED:

```
A>ED B:SAMPLE.ED   (ED Activates, SAMPLE.ED opens)
:*I                (Establishes Insert Mode)
1: *THIS EXAMPLE SHOWS HOW TO
2: *ENTER TEXT
3: *USING THE ED UTILITY
4: *^Z             (Returns to ED command prompt)

:*E                (Saves the file and returns to A>)
A>
```

Editing Commands

The list below summarizes **ED** line editing commands. The *Osborne Executive Reference Guide* has more information about each of them.

Some of these commands include the letter *n*. Replace *n* with a number when using the command. Positive numbers move the pointer forward through the file. Negative numbers move it backward. Using the **#** symbol rather than a number activates the command from the current line to the end of the file.

Text Moving Commands

- nA*** copies next *n* lines of the file to the buffer.
- E*** saves edited text and returns to the **A>** prompt.
- H*** saves edited text and returns to the ED prompt **:***.
- nW*** writes first *n* lines of the memory buffer to the free space in the temporary file.

Pointer Moving Commands

- B*** moves the pointer to the first line in the file.
- B*** moves the pointer to one line beyond the last line in the file.
- nC*** moves *n* characters to the right (positive number) or left (negative number) of the current character.
- nL*** moves the pointer *n* lines forward (positive number) or back (negative number).
- nT*** displays *n* lines before or after the current one. A negative number displays lines before the current one. A positive number displays the current line and those following.
- n*** moves the pointer forward or back *n* lines and displays the *n*th line.
- n:*** moves the pointer to line number *n*.

Text Changing Commands

- nD*** deletes *n* characters before or after the current pointer position.
- Fstring^Z*** finds the specified string. Format:
Fstring^Z

- nK** deletes **n** lines before or after the current one.
- S** replaces characters on the current line with new ones. Format:
Soldstring^Znewstring^Z

Insert Commands

- I** establishes Insert Mode.
- ^Z** returns from Insert Mode to ED prompt :*.

Section 6: File Processing and Data Handling

<i>SUBMIT</i> <i>PUT</i>	<i>GET</i>
-----------------------------	------------

CP/M Plus has several utilities for executing sequential commands and managing stored data.

SUBMIT executes CP/M Plus commands contained within a file. These commands are carried out as if they were individual instruction lines in a program.

PUT directs data between the computer and devices connected to it. You can use it, for example, for file storage of data usually directed to a printer, console or other device.

GET is a utility for retrieving stored data from a file and using it in the file or program you're currently working with.

This section describes how to use each of these utilities.

File Processing with SUBMIT

You may find that certain command sequences recur as you use the Osborne Executive. The commands for time stamping a diskette and displaying its directory are an example. You can create a **file of these commands** called a Submit file, then use **SUBMIT** to carry them out automatically.

These are the guidelines for Submit files:

1. Use WordStar or **ED** to create the file.
2. Include the file type **.SUB** after the file name.
3. Include only commands which are valid at the **A>** prompt level.

A Submit File Example

In the following example, we'll use **ED** to create a Submit file named **SETUP.SUB**. We'll include in this file commands for time stamping and directory display. To create **SETUP.SUB**, type the following:

```
A>ED B:SETUP.SUB
:*|
1:*DATE SET
2:*INITDIR B:
3:*SET [CREATE=ON]
4:*SET [UPDATE=ON]
5:*SETDEF [DISPLAY]
6:*DIR [FULL]
7:*^Z
:*E
A>
```

The commands in this file will automatically:

- set the date
- initialize the diskette in drive **B**
- turn **ON** the **CREATE** time stamp option
- turn **ON** the **UPDATE** time stamp option
- display the name and location of each program executed
- display a complete directory for drive **A**

To execute the **SETUP.SUB** file for drive **B**, type

```
A> S U B M I T SPACE BAR B : S E T U P
```

and press **RETURN**.

SUBMIT carries out the sequence of commands quickly and efficiently. Here's the general format for executing a Submit file. Specify the disk drive if it isn't the one currently logged:

```
A> SUBMIT {filename}
```

If you enter a **SUBMIT** command **without including a file name**, you'll see this message in response:

Enter File to Submit:

Type the file name at this point and press **RETURN** to execute the file.

Multiple Submit Files

A Submit file contains a sequence of commands or instructions. Not only can you create several Submit files, you can also include or "nest" these files in each other. Executing a Submit file automatically carries out any Submit files nested in it.

The following example shows the file **COMPILE.SUB** nested in the file **COMPALL.SUB**.

COMPILE.SUB contains

```
RMAC $1  
LINK $1
```

COMPALL.SUB contains

```
SUBMIT COMPILE ACCOUNT1
SUBMIT COMPILE ACCOUNT2
SUBMIT COMPILE ACCOUNT3
DIR
```

The instruction for executing **COMPALL.SUB** is:

```
A>SUBMIT COMPALL
```

The system would then:

- read the first line in **COMPALL.SUB**
- locate the **COMPILE.SUB** file
- substitute **ACCOUNT1** for **\$1** in the file
- carry out the instructions in **COMPILE.SUB**

The remaining lines in **COMPALL.SUB** would be read and carried out in sequence. At the fourth line, the system would display the directory and return to the **A>** prompt.

The utilities described in the rest of this chapter are on the CP/M Advanced Utilities Disk. Place a working copy of this disk in drive **A** and press **RETURN**.

The PUT Command

The **PUT** command directs output from a file to another destination or back again. Its general format is:

```
A>PUT {destination1} OUTPUT TO {destination2}
{[options]}
```

The following example routes console output of a program named ACCOUNT3 to the JONES file in drive B:

```
A>PUT CONSOLE OUTPUT TO FILE B:JONES
```

```
A>TYPE ACCOUNT3
```

The output of ACCOUNT3, normally sent to the video monitor, would now be sent to the JONES file instead.

PUT Command Options

The PUT command has several options for displaying output, filtering control commands, or using the SYSTEM feature.

The format for PUT options is to include them in brackets at the end of the PUT command. Here's an example:

```
A>PUT CONSOLE OUTPUT TO FILE NEXFILE [NO ECHO]  
A>TYPE MYPROG
```

This command directs console output of the file MYPROG to the file NEXFILE *without displaying the file contents on screen*. Output to NEXFILE stops automatically at the end of MYPROG.

The PUT options are:

[ECHO]	default setting. Displays console output as it's directed to a file.
[NO ECHO]	prevents screen display during output.
[FILTER]	translates control commands into printable characters.
[NO FILTER]	prevents translation of control characters to printable form, for virtual image of screen handling.
[SYSTEM]	directs system and program output from a specified device to a specified file. Use an additional PUT command to restore output to its usual destination.

The following example directs printer output to the printer and to the NEWACCT file. It also displays the output on screen:

```
A>PUT PRINTER OUTPUT TO FILE NEWACCT  
    [ECHO,SYSTEM]  
A>TYPE MYPROG
```

Because the SYSTEM option is in effect, output will continue to the file and printer until you enter another PUT command.

A PUT command containing the SYSTEM option must be terminated with another PUT command. The general format for this terminating command is:

```
A>PUT {destination1} OUTPUT TO {destination1}
```

The following command, for example, restores console output to the console:

A>PUT CONSOLE OUTPUT TO CONSOLE

If drive **B** is the default drive, enter the terminating command after the **B>** prompt.

The GET Command

GET substitutes file input for console input. For example, if one program's output is another's input, **GET** will direct the system to the appropriate input file. It then makes this input available for the next system command or program.

This is the **GET** command format:

A>GET FILE {filename} {[options]}

GET has three options:

- | | |
|------------------|--|
| [ECHO] | the default setting. Displays input at the console. |
| [NO ECHO] | prevents input display at the console. Displays output and system prompts. |
| [SYSTEM] | begins console input from the specified file without waiting for program to run. |

If the **SYSTEM** option isn't included, one system command can initiate a user program. The program's console input comes from the file specified in the **GET** command. **GET** takes system and program input from the file until the file ends or it finds a **GET CONSOLE** command in the file.

In the following example, GET executes the user program named PROGRESS with the file INVENT as input:

```
A>GET FILE INVENT  
A>PROGRESS
```

Since we didn't use the SYSTEM option, we had to specify a user program from the console. When PROGRESS terminates or the system finds a GET CONSOLE command in the INVENT file, it reverts to the console for the next input.

We'd use the SYSTEM option in our example by typing

```
A>GET FILE INVENT [SYSTEM]
```

In this case, the system would immediately go to the INVENT file for its console input. If INVENT contains a GET CONSOLE command, control reverts to the console at that point. Otherwise, INVENT provides all console inputs until the file ends.

The command which restores control of the keyboard or other console input device is:

```
A>GET CONSOLE
```

Section 7: Running Assembly Language Programs

<i>MAC</i>	<i>LINK</i>
<i>RMAC</i>	<i>SID</i>

This section reviews CP/M Plus utilities for assembly language programming. These utilities create, assemble, load, and execute programs in 8080 assembly language. Familiarity with 8080 assembly language programming is a prerequisite for this section.

MAC and RMAC Assemblers

MAC and **RMAC** are CP/M Plus assemblers which translate assembly language code into hexadecimal or relocatable object code.

MAC is an assembler which produces absolute object code. This code does not require a **LINK** step.

RMAC produces relocatable object code which is easier to move in memory than absolute code.

These assemblers generate several files:

- a **.HEX** file (generated by **MAC**) of hexadecimal code or a **.REL** file (generated by **RMAC**) of relocatable code.
- a **.SYM** file containing a sorted list of symbols defined in the program.

- a **.PRN** file containing an annotated source listing which can be printed or examined at the console.

In this section, we'll use **RMAC** as our assembler for a sample program called **COINFLIP**.

Assembling a Program

Insert the CP/M Plus Advanced Utilities Disk in drive **A** and a formatted work diskette in drive **B**. Use the **ED** utility or WordStar non-document mode to create the file **COINFLIP.ASM**. You must include the **.ASM** file type in files for assembly language code.

We'll write the sample assembly language code in this file. Enter it as follows:

```

true      equ  (1=1)      ;Define value for true conditionals
false     equ  not true   ;Define value for false conditionals

WBoot Entry equ 00000h    ;Warm boot entry point
BDOS      equ 00005h     ;BDOS entry point
TPABase   equ 00100h     ;Base of CP/M's Transient Program Area

DirectConIO equ 006h     ;BDOS direct console I/O function
CharRequest equ 0FFh     ;Character request code for direct console I/O
PrintString equ 009h     ;BDOS print string function

cr        equ 00Dh       ;ASCII carriage return
lf        equ 00Ah       ;ASCII line-feed
CTRLC    equ 'C'-040h   ;ASCII CTRL/C (ETX)

RMAC      equ true       ;Assembly toggle for RMAC
          equ not RMAC   ;Set RMAC to false if MAC is to be used
MAC       equ not RMAC   ;Assembly toggle for MAC

          if MAC         ;Org at 100h only if assembled with MAC
          ;since linker automatically inserts a 100 byte
          ;offset.
          org TPABase    ;Start code at base of TPA
          endif

```

```

lxi SP,StackArea ;Initialize stack pointer to point to local
                    ;stack

lxi D,InstrMessg ;Call BDOS to print instructions. DE > String
mvi C,PrintString ;C = BDOS function #
call BDOS ;

```

Flop:

```

inr B ;Increment counter

push B ;Save counter around call to BDOS
mvi E,CharRequest ;Ask BDOS if there is a char from the keybd
mvi C,DirectConIO ;
call BDOS ;
pop B ;Restore Counter

ora A ;Is ther a char? (I.E., is A not zero?)
jz Flop ;No char ready, loop again

cpi CTRLC ;Is it CTRL/C?
jz WBootEntry ;Yes, return to CP/M through warm boot entry

mov A,B ;Put counter value in REG A
ani 0th ;Set zero flag if low order bit is zero (I.E.,
        ;counter is even.)
jz Heads ;either way.)

```

Tails:

```

                    ;Counter is odd, which means tails.
lxi D,TailsMessage ;Point DE at tails message

```

Heads:

```

lxi D,HeadsMessage ;Point DE at heads message
jmp PrintMessage ;and fall through to Print Message routine:

```

PrintMessage:

```

mvi C,PrintString ;Tell BDOS to print the message
call BDOS ;
jmp Flop ;and do it again

```

InstrMessg:

```

db 'This program simulates a coin flip. Every time you '
db 'press a day, the' ,cr, lf
db 'Executive will print out that the flip was heads or '
db 'tails.' ,cr,lf
db 'Press CTRL/C to return to the A> prompt.' ,cr,lf, '$'
;String is terminated with a $ so CP/M knows when to stop
;printing chars

```

```
TailsMessage:
db 'The flip is tails.' ,cr,lf, '$'

Heads Message:
db 'The flip is heads.' , cr,lf, '$'

db 30*2 ;Set up 30 level stack
StackArea equ $ ;Set StackArea label to point to top of stack
```

After entering the code, the next step is to assemble the program. Here's the general format of the **RMAC** utility we'll use:

A>RMAC filename{.filetype} {\$options}

On this command line, **filename** is the name of the file being assembled. Its file type *must be* ASM. **\$options** designates RMAC command options. Precede the first option with a dollar sign (\$) and separate the options with a space.

These are the RMAC options.

RMAC Command Options

Rr	specifies destination drive for REL file (A-O, Z)
Sr	specifies destination drive for SYM file (A-O, X,P,Z)
Pr	specifies destination drive for PRN file (A-O, X,P,Z)

The letter **r** in each option is a variable. You can specify it as follows:

A-O	designates disk drives
X	designates console
P	designates printer
Z	designates no output files created

We'll specify options to assemble COINFLIP and create three output files in drive B. Type:

```
A> R M A C SPACE BAR B : C O I N F L I P . A S M  
SPACE BAR $ R B SPACE BAR S B SPACE BAR P B
```

and press RETURN.

Drive B now has three output files: COINFLIP.REL, COINFLIP.SYM, and COINFLIP.PRN. To display the hex code, type:

```
A> T Y P E SPACE BAR B : C O I N F L I P . P R N
```

and press RETURN.

We could have assembled COINFLIP with MAC rather than RMAC. The format for MAC commands is:

```
A>MAC filename {$options}
```

Except for Rr, MAC has the same options as RMAC. Use H (HEX file) rather than R (REL file).

Creating a Cross-Reference File

When using long programs, it's convenient to have a list of variables and their occurrence in the program. XREF provides such a reference file. The XREF utility has this general format:

```
A>XREF filename {$Pp Ss}
```

Filename is the name of the assembled program file. Precede the first option with a dollar sign (\$) and separate each option with a space. The options specify PRN and SYM file locations if they're not on the default drive. XREF uses the SYM copy to make the reference file.

To create the reference file for COINFLIP, type:

```
A> X R E F SPACE BAR B : C O I N F L I P
      SPACE BAR $ SPACE BAR P B SPACE BAR S B
```

and press **RETURN**. The options **PB** and **SB** indicate that the input files are on drive **B**.

To see a sample reference file type:

```
A> T Y P E SPACE BAR B : C O I N F L I P . X R E F
```

and press **RETURN**.

```
CHARREQUEST
CR          00FF    10#  38
            000D    13#  70  72  73  78  81

Press RETURN to Continue
CTRLC      0003    15#  47
DIRECTCOMIO 0006    9#  39
FALSE      0000    3#
FLOP       000B   34#  44  66
HEADS      002A   53  60#
HEADSMESSAGE 00F9   60  80#
INSTRMESSG 0038   30  68#
LF         000A   14#  70  72  73  78  81
MAC        0000   19#  21
PRINTMESSAGE 0030   58  61  63#
PRINTSTRING 0009   11#  31  64
RMAC       FFFF   17#  19
STACKAREA  014A   27  84#
TAILS      0024   56#
TAILSMESSAGE 00E4   57  77#
TPABASE    0100    7#  24
TRUE       FFFF    2#   3  17
MBOOTENTRY 0000    5#  48

A> █
```

Displaying Relocatable Object Code

The **DUMP** command displays relocatable object code created by RMAC. To display relocatable hex code for COINFLIP, type:

```
D U M P SPACE BAR B : C O I N F L I P . R E L
```

and press **RETURN**. The first portion of the output looks like this:

```
CP/M 3 DUMP - Version 3.0
0000: 85 90 D3 D2 53 91 93 25 00 00 13 54 A0 11 8D 4A   ....S..X...T...J
0010: 01 08 D3 80 00 70 25 9A 05 00 01 18 A1 E7 F8 38   .....pX.....8
0020: 0C CD 02 80 18 2B 76 55 0B 00 7F 00 D9 40 00 01   .....+vU.....@
0030: E1 CC 01 65 52 A0 00 BD E4 00 61 D3 00 00 8D F9   ...eR.....a....
0040: 00 61 D3 00 00 70 25 9A 85 00 30 E8 58 01 50 D0   .a...pX...0.X.P.
0050: 69 39 88 0E 07 23 79 9C E4 61 36 88 0E 66 93 69   i9...#y..a6..f.i
0060: D4 D8 61 3A 19 4E 62 03 08 80 C6 6F 34 9B 84 06   ..a.:Hb....o4...
0070: 63 61 A4 E0 2E 10 11 4E C6 53 91 E4 40 74 34 9B   ca....W.S..@t4.
0080: 4C A2 03 C9 BC EA 20 38 1C 8C A7 33 98 80 C2 20   L.....8...3...
0090: 35 99 4F 22 C1 01 D0 D0 65 06 82 88 A7 83 29 8C   5.0'.....e.....)
00A0: EA 74 34 9D 8C A2 03 B9 A4 D8 6C 10 1C 0E 46 93   .t4.....1...F.
00B0: 71 D0 40 6F 3A 9D 04 07 43 41 84 E8 20 3A 1A 0C   q.@o:...CA... ..
00C0: A2 03 31 B0 D2 70 10 1D CC 27 31 01 A0 CA 61 32   ..1..p...'1...a2
00D0: 1C C4 06 F3 90 80 E8 61 34 9B 0E 62 E0 68 28 A0   .....a4..b.h(.
00E0: 72 32 9C CE 62 02 19 50 A4 4C 17 90 C4 07 43 78   r2..b..P.L....Cx
00F0: 80 E4 65 3A 1D 4E 46 E1 01 D0 DE 20 3A 1A 0C A2   ..e.:HF..... :...
0100: 02 08 F8 40 70 39 1B CD A7 03 A0 B8 1B 0A 12 15   ...@p9.....
0110: 0D 06 51 01 98 D8 69 38 08 0D 27 31 01 D0 C2 69   ..Q...i8..'1...i
0120: 36 1C C5 C0 D0 50 90 A8 68 32 88 0C C6 C3 49 C0   6....P..h2....I.
0130: 48 69 39 88 0D 06 53 09 90 E6 2E 06 82 84 92 D4   @i9...$.
0140: A0 19 C0 00 00 9E 1A   .....
0150: 1A   .....
Press RETURN to continue █
```

Enter **CTRL C** to terminate the DUMP display.

Preparing a File for Execution

LINK converts program code from relocatable to executable form so the program can be run. **LINK** creates a **.COM** file from a **.REL** file. It can also combine several **.REL** files into one large **.COM** file or include referenced modules in the output file.

This is a typical **LINK** command for combining files:

```
A> LINK B:ACCOUNTX,STATFILE,TABLE
```

This command combines three **.REL** files (**ACCOUNTX**, **STATFILE**, and **TABLE**) into one **.COM** file named **ACCOUNTX**. The first name specified in the **LINK** command becomes the name of the combined file.

LINK also resolves the external references in each of the files. It displays the new **.SYM** file as a symbol table on screen. This file contains the external references defined in the linked **.REL** files.

LINK displays error messages before displaying the contents of the new **.SYM** file. We suggest you use **PUT** to direct the console input to diskette. This is a convenient way to have the error messages at hand as you review the **.SYM** file.

The object code for programs assembled with **MAC** has a **.HEX** file type. The *Osborne Executive Reference Guide* describes the **HEXCOM** command which produces **.COM** files from **.REL** files.

The following **LINK** command creates a **COINFLIP.COM** file from **COINFLIP.REL**, making **COINFLIP** an executable program:

```
A> LINK B:COINFLIP
```

Running a Program

To run COINFLIP, type:

```
A> B C O I N F L I P
```

and press RETURN. If the program is running correctly, pressing any key will display either HEADS or TAILS and then the A> prompt. If this doesn't happen, check that you've copied COINFLIP correctly. Use ED or WordStar to edit it if necessary.

Modifying Programs with SID

Checking the ASM file visually is the most efficient way to modify a short program like COINFLIP. After modifying the program, edit it, use RMAC and LINK to assemble it, then run the program again.

SID—the Symbolic Instruction Debugger—will modify or debug more intricate programs. **SID** can:

- execute a program
- interrupt it at any point
- change instructions
- execute individual instructions

As an example of SID's use, we'll change COINFLIP from a HEADS/TAILS coin-flipping program to a BUY/SELL program. The method we use here changes COINFLIP in memory but not on diskette.

The first step is to activate **SID**. Type:

A> **S I D** **SPACE BAR** **B . C O I N F L I P . C O M**

and press **RETURN**. The system responds with **SID's #** prompt. This prompt requests an address. Type:

D 0 1 8 0

and press **RETURN**. This displays a page of memory beginning at hex 0180. It looks like this on screen:

```
#
#D0180
0180: 63 75 74 69 76 65 20 77 69 6C 6C 20 70 72 69 6E cutive will prin
0190: 74 20 6F 75 74 20 74 68 61 74 20 74 68 65 20 66 t out that the f
01A0: 6C 69 70 20 77 61 73 20 68 65 61 64 73 20 6F 72 lip was heads or
01B0: 20 74 61 69 6C 73 2E 0D 0A 50 72 65 73 73 20 43 tails...Press C
01C0: 54 52 4C 2F 43 20 74 6F 20 72 65 74 75 72 6E 20 TRL/C to return
01D0: 74 6F 20 74 68 65 20 41 3E 20 70 72 6F 6D 70 74 to the A> prompt
01E0: 2E 0D 0A 24 54 68 65 20 66 6C 69 70 20 69 73 20 ...$The flip is
01F0: 74 61 69 6C 73 2E 0D 0A 24 54 68 65 20 66 6C 69 tails...$The fli
0200: 70 20 69 73 20 68 65 61 64 73 2E 0D 0A 24 00 00 p is heads...$.
0210: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0220: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
#
```

We'll change the output messages from **HEADS/TAILS** to **BUY/SELL**. Location 0200 contains the **H** of **HEADS**. We'll start to modify the code here. Type:

S 0 2 0 0

and press **RETURN**. **SID** responds with the contents of address 0200H. Use **RETURN** to index down (over) six places to 68.

The cursor is at 74 which is hex for the "T" in TAILS. Beside this 74, type:

`"SEL$ RETURN`

When you've made the change, type period (.) to leave code-modification mode. The system displays **SID's #** prompt.

To run the BUY/SELL version of COINFLIP while still in SID, type:

`G100`

and press `RETURN`.

While the program's running, press any key to get a BUY or SELL decision.

When you're ready to return to CP/M Plus, enter `CTRL C`.

The procedure we used didn't affect the diskette copy of COINFLIP. On diskette, it's still a HEADS/TAILS coin-flipping program. The *Osborne Executive Reference Guide* describes how to change the program permanently using SID.

PART 1
p-System

Introduction

The p-System is an alternative operating system to CP/M Plus. With the p-System, you can use languages and programs written for the p-System on your Osborne Executive.

The p-System guide is divided into three sections:

- a tutorial covering file manipulation and program execution
- a description of the p-System Editor
- and a description of the PRINT utility

SECTION 1
Tutorial

Learning to Use the p-System

This chapter describes the p-System, an alternative operating system to CP/M Plus. When you use the p-System instead of CP/M Plus, you can use programs and computer languages written for it rather than CP/M Plus.

The p-System Operating System

The p-System supplied with the Osborne Executive is part of a larger one called the *Development System*. The p-System you've received has utilities to copy and execute program code from programs written in UCSD Pascal. You can also produce documents with the Editor and print them with the PRINT utility.

The Development System

The Development System is the entire group of programs that allows you to develop programs. Osborne will soon sell this group of programs in the Osborne Approved Software program. The Development System includes such "utility" programs as the Filer (also included on your p-System diskette), which allows you to manipulate individual files of instructions or data; the Editor (included on your diskette), with which you create or edit files of instructions or data; the Compiler, which turns the instructions you type into executable programs; and a number of other smaller, miscellaneous modules.

One of the programs included in the full Development System is the UCSD Pascal compiler. You use this compiler to convert your instructions into executable computer code. The programs you run using the p-System were written in UCSD Pascal.

Pascal is not the only language available on the Development System; BASIC and FORTRAN are also available, as well as an assembler. Programs written in UCSD PASCAL, BASIC, and FORTRAN all create p-Code. You can link together modules of code written in these languages. The p-Code for each of the languages are the same.

To summarize, the p-System you received allows you to manipulate files, execute programs, and edit and print text files. The full Development system, which will be available soon from Osborne, allows you to develop your own programs.

What's Included With Your p-System

Your system includes these programs and utilities.

The **FILER**: allows you to manipulate files with these options:

LDIR	Lists the directory
REM	Removes files
CHNG	Renames files
TRANS	Copies files
DATE	Sets the date and time
QUIT	Quits the Filer
BAD BLKS	Finds bad blocks
EXT-DIR	Lists an extended directory

KRNCH	Moves files together and consolidates disk space
MAKE	Creates blank files and recovers files
PREFIX	Sets reference method to devices
VOLS	Lists devices
XAMINE	Recovers data in bad blocks
ZERO	Initializes the directory
GET	Gets a work file
WHAT	Lists the current work file
NEW	Clears the work file
SAVE	Saves a work file with a new name

XECUTE: allows you to execute programs, such as:

MARKDUPDIR	Creates a duplicate directory
COPYDUPDIR	Copies a duplicate directory
RECOVER	Recreates directories and files
UTIL	A program that contains these options:
FORMAT	Formats a diskette
MAKE BOOT	Writes boot tracks
CONFIGURE	Configures boot tracks
BACK-UP	Creates duplicate diskettes
PRINT	Formats and prints text files

The **HALT** option, which halts execution and exits the user from the p-System.

The **INITIALIZE** option, which restarts the system.

The **USERRESTART** option, which reruns the last program used.

The **MONITOR** option, which places p-System commands and options that you type in an executable file.

The **EDITOR** option, which allows you to create and edit text files.

As you read through the lessons that follow, use your Executive to duplicate the interaction you see. You will be amazed at how easy it is to use the p-System.

Organization of This Manual

The documentation on the p-System is divided into three parts: A tutorial covering file manipulation and program execution, a description of the Editor, and a description of the PRINT utility. We suggest that you read the first section (on file manipulation and program execution) before you read about the Editor and the PRINT utility.

Preparing to Use the p-System

In this section we'll describe p-System's prompts and terminology.

Copying Your Master p-System Diskette

Copy your master p-System diskette before you do anything else.

Use the CP/M Plus Copy command to copy your master p-System diskette as described in Chapter 1. Once copied, the master diskette should be safely stored away from electrical and magnetic devices.

Be sure you label your copy.

Later, we will tell you how to use the p-System to create and copy diskettes.

Loading the p-System

Load the p-System as you do any system diskette. Turn the machine on or press RESET, place your copy of the p-System diskette in Drive A and press **RETURN**. You will notice that it takes quite a bit longer to load p-System than to load CP/M Plus. That is because the p-System is a much larger system. It not only does a few things that CP/M Plus does not do, many utilities that are provided on a separate diskette with CP/M Plus are on the p-System's system diskette.

When the system is loaded, you see a promptline and a message on the screen.

```
Command: E(dit, R(un, F(ile, C(omp, L(ink, X(ecute A(ssem, D(ebug, ? [IV.12, B]
Welcome OSBORNE, to
      U.C.S.D. p-System IV.12 B
      Current date is 10-Mar-83
```

The p-System's Promptline

Most computer systems show you a "system prompt" and then wait for you to type a complete command. For example, the CP/M Plus prompt is usually **A>** after which you type a command, such as **DIR**, and then press RETURN.

With the p-System, the concept of prompting the user for a command changes. Instead of the simple prompt mentioned above, you see a more complex prompt, normally called the *promptline or menu*.

```
Command: E(dit, R(un, F(ile, C(omp, L(ink, X(ecute A(ssem, D(ebug, ? [IV.12 B]
```

This promptline offers you a choice of things to do. The p-System expects you to press a letter to indicate which of the possibilities you wish to perform.

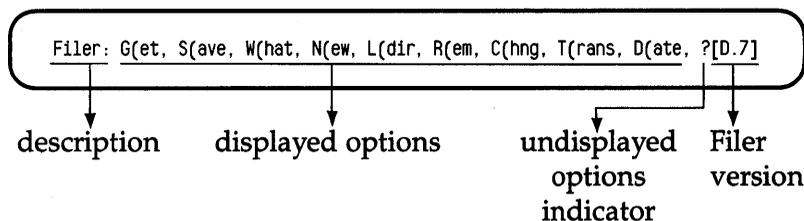
For instance, if the main promptline shown above appears on your screen, you can press the letter **F** to tell the system that you wish to use the **Filer**. You need not press the **RETURN** key to verify your choice—when you press the **F** key, your request is accepted and acted upon immediately.

Press **[F]**.

The promptline concept is really rather simple. Each promptline consists of three basic components:

Description
Options
Version/Other Options

Let's look at the promptline you now have on your screen and separate out the three basic components.



The description on a promptline comes first. In our example the description is *Filer*:. In most cases the description ends in a colon, but some utilities may not use this convention.

What the description does is tell you which major module (program, or section of the p-System) you are using. If you are using the *Filer*, the promptline reads **Filer**; if you are at the highest command level (as when starting), you'll see **Command**..

After the description comes a list of commands or options available to you. The capital letter that precedes each "(" is the

letter to type to execute that option. The full option name — at least the full option name as it appears on the screen — consists of a capital letter, a left parenthesis, and the remainder of the option name. Here is part of the list of options that appear in the Filer promptline we're studying:

Option on Screen	Press Key	What it Does
G(et	G	Gets a file for use
S(ave	S	Saves the current workfile
W(hat	W	Identifies the current workfile
N(ew	N	Clears the current workfile
L(dir	L	Lists a directory of files

As you can see, the option names—with possibly the exception of **L(dir**—suggest the action that is about to take place (actually, even **L(dir** makes sense if you simply remember it is short for “list directory”).

We'll find the use of a capital letter followed by a left parenthesis a common practice throughout the p-System, not just on the promptline, so before moving on, make sure you understand the concept of separating the key to press from the rest of the name of the command (with the parenthesis).

The last part of the promptline consists of a question mark (sometimes) and a version number enclosed within square brackets at the end of the line.

The question mark is simple to explain: if it appears, there are more prompts possible than can fit on the current line. There are three promptlines in the **Filer**.

Press the **[?]** key to display the other options available. Continue to press **[?]** until you have seen all the **Filer** promptlines.

Just because an option doesn't show on the top line of the screen doesn't mean that the option you specify won't be recognized. For instance, if the first **Filer** prompt we showed you appeared on your screen, you could choose an option from the second promptline. Pressing ? is simply a way of showing you the other options that are available. If no question mark appears, then there are no more options. Now, let's go back to the Command promptline by pressing **Q** for Quit.

The numbers and letters that appear within the square brackets refer to the version of the current program you are using. The **[IV.12 B]** that appears on the first promptline when you use the p-System (the Command promptline), indicates that you are using version IV.12 B of the p-System. We suggest that you develop the habit of identifying your programs in the same manner, as this is a simple and effective method for showing the "vintage" of a program. If you later receive an update to your software and the promptline shows the new version number, this is your verification that you are using the correct version of the operating system, and haven't accidentally picked up an older version.

We could make our explanation of getting around in the p-System much more elaborate, but unless you are using the Development System, it isn't necessary to go into greater detail than is presented here and in the next lessons.

With all that preliminary information taken care of, you are ready to learn about using the p-System.

Starting a Session and Preparing a Blank Diskette

In this lesson you'll learn how to start a p-System session and how to prepare a blank diskette for use.

You still have the **Command:** promptline on the screen:

```
Command: E(dit, R(un, F(ile, C(omp, L(ink, X(ecute, A(ssem, D(ebug, ?[IV.12 B]
```

But what then?

Setting the Date

We recommend that the first thing you do during any session with the p-System is set the date. If you do not do this, you lose a potentially useful piece of information about your files.

The p-System keeps track of the date when you last changed a file. Obviously, then, you need some way of telling the system the date.

To gain access to the **Date** option, press the **[F]** key to get the **Filer:** promptline (the **Date** option is kept in the **Filer**.)

```
Filer: G(et, S(ave, W(hat, N(ew, L(dir, R(em, C(hng, T(rans, D(ate, ? [D.7]
```

Press the letter **Q**. Now you should see the following display.

```
Date set: <1..31>--<JAN..DEC>--<00..99>
```

```
Today is 10-Mar-83
```

```
New date?
```

The computer is waiting for you to enter a new date in the format indicated by the top prompt line (i.e., the day, a hyphen, the first three letters of the month, a hyphen, and the last two digits of the year.) If today is the third day of May, 1983, you would type:

```
3 - M A Y - 8 3
```

followed by **RETURN** to indicate to the system that you are finished.

Right now, enter today's date.

Once you've typed the date and pressed **RETURN**, the computer indicates the date you set. If you make a formatting mistake, the p-System uses the original date it had, not your new, incorrect one. The **Date** option accepts information *only* in the format specified above. Any other format results in no new date being set.

Alternatively, you can type just the day — for example, 8 for the eighth of January — if the month and year are as you want them. Or you can enter just the day and the month (separated by a hyphen) if they are all that have to be changed.

When you are finished, the p-System returns you to the **Filer**: promptline.

Press **Q** for **Quit** to return to the Command level.

Formatting a Blank Diskette

As you know, all of your diskettes have to be formatted before they can be used. You can use the CP/M Plus Copy command to do this, but you can also use the p-System's formatter—and since you already have the p-System diskette in Drive A, why not?

Place a blank, unformatted diskette in Drive B.

You have the command line on your screen:

```
Command: E(dit, R(un, F(ile, C(omp, L(ink, X(ecute, A(sssem, D(ebug? [IV.12 B]
```

Press **X** for X(ecute. The system now asks:

```
Execute what file?
```

Type **U T I L** and press **RETURN**. Soon you will see the UTIL promptline:

```
UTIL: F(ormat B(ack-up C(onfig M(ake boot Q(uit
```

The screen also tells you that "Pressing the escape key at any prompt will return you to this level."

Press **F** for **Format** to format a diskette. You are now asked for the unit number to be formatted.

```
Disk Format Utility
```

```
Enter unit number of disk to be formatted (4,5)
```

Type **5** followed by **RETURN**. The p-System refers to diskette locations as *volumes*. What is called drive B in CP/M Plus is referred to as volume #5 in the p-System. Drive A is referred to as volume #4. If you change your mind about formatting anything at all, press **ESC** to exit without formatting.

Now the system asks:

OSBORNE format or Universal format? (O/U)

Type **O** for now. U stands for Universal Medium format, which is another format that the p-System can use. It is discussed later in this chapter.

WARNING: Formatting will destroy all data on the diskette.

The Osborne format option creates a blank, double-density Executive format diskette. This formatted diskette has 390 blocks with 512 bytes per block.

The system displays:

Place disk to be formatted in drive #5
and press RETURN when ready

Do that now if you have not done so already. Press **RETURN**.

When the formatting stops, press **Q** to return to the **Command:** promptline.

The next thing to do is to create a directory for your newly formatted diskette.

Creating a New Directory

The **Filer** option, **Zero**, creates the initial directory on a new diskette.

The **Command:** promptline is on the screen, so press the **[F]** key to display the **Filer:** promptline.

Now press **[Z]**, for **Zero**, to “zero” a new directory. The computer asks you several questions:

Zero dir of what vol ?

Type **[#][5]** in response to this question and press **[RETURN]**.

If this were a diskette that was not freshly formatted, you would be using the **Zero** option to reinitialize the directory. When you reinitialize a directory, it destroys the contents of the diskette and frees it for new files.

Now the system asks:

Duplicate dir?

Type **[N]** for no. You would type **Y** if you wanted to maintain a backup directory, which is a good idea as you will learn later. Right now, however, we have you answer **N** because we want to show you another method for creating a backup directory later in this chapter.

Now your screen displays more questions:

of blocks on the disk?

Type **3** **9** **0** and press **RETURN**. Because this diskette is new and hasn't been used previously, you need to type in the number of blocks to use. For Executive double-density diskettes, that number is normally 390, but you can specify less. You cannot specify more than 390. Whatever number you type, this is an irreversible decision. Once files are stored on the diskette, you must use **Zero** to reinitialize with another block size, thus destroying the directory.

If the diskette has been used by the p-System and has not been reformatted, zero reports how many blocks it thinks are on the diskette.

New vol name?

Here's where you get to specify the volume name — a name for the diskette. We named our p-System diskette **OSBORNE:** so that you would know that it was the original from Osborne. We suggest you invent a naming pattern that is meaningful to *you*. No need to be modest; name this diskette after yourself. The name can be no more than seven characters long. Do not type a colon at the end of the name, zero will add this automatically. After you have typed the volume name, press **RETURN**.

MYDISK: correct?

Just to make sure that you didn't mistype the volume name, the p-System asks you to verify the name you just input. As before, press **Y** to verify the action, **N** to cancel it.

Press **Y**.

(You have named your diskette after yourself. When we talk about this diskette, we'll call it MYDISK.)

When you are finished, you are automatically returned to the **Filer:** promptline. Your newly formatted and zeroed diskette is now ready for use.

Now you are almost ready to learn about file handling. First you need to know the p-System's file-naming rules. And we're sure that, by now, you feel a need to understand a little more about volumes. There isn't much "hands on" work in the next lesson, but the information there is vitally important. So take a break, then sit back and carefully read the next lesson before you go on to more work at the keyboard.

Understanding File Names, File Types, and Volumes

This lesson explains the p-System's rules for naming files and for referring to volumes — the devices that send, receive, or store your files.

This is a reading lesson, primarily. You will have a chance to use what you learn here in the next lesson. So read carefully. No need to memorize; just make sure you understand the concepts. You can always come back to this lesson if specific details slip your mind.

File Names

File names may consist of up to 15 characters. Any lowercase character you type for a file name is automatically converted to uppercase by the computer. If you include a blank in a file name, the p-System will ignore it; the same holds true for control characters.

In addition to the standard alphabet, file names may also contain several special characters. The full list of characters that may appear in a file name are as follows:

A B C D E F G H I J
K L M N O P Q R S T
U V W X Y Z / - _ .
1 2 3 4 5 6 7 8 9 0

File names can never contain the following characters:

\$: = ? ,

If you attempt to name a file using one of these special characters, the p-System may seem to recognize it, but the file name that is used by the system most likely will not be the same as the one you intended. The reason for this will become clear as you learn about some of the other aspects of the p-System. For now, just realize that typing \$, :, =, ?, or , causes the system to assume something about your file name that you might not have intended.

Some examples of valid file names are:

SYSTEM.WRK.TEXT
OSBORNE/ONE.WRK
THIS.WAY.AND.TH
PROG-2.CODE

A final comment about file names is in order. Make sure that the name is meaningful. You have 15 characters, so you should be able to say a lot with a file name. For example, you can name a file **AP12**. That doesn't mean much. A better name would be **PAYABLES.12**. But, you might think that the 12 means the twelfth file or the twelfth month. **PAYABLES.JAN.2** can leave no doubt; this name correctly indicates the file's contents.

To summarize, file names should clearly identify the contents of a diskette file, and can be up to 15 characters long.

File Types

File names may contain special suffixes. These suffixes usually indicate the *file type*. Since files can perform so many different functions within the p-System, we need some method of telling the computer what the contents of a file are to be used for. To do so, we use the last part of the file name for the file type.

File types make it easier for the system to assume information about your files and what they are to be used for. If all files

had the same file type, the computer might not know when to interpret a file as instructions and when to interpret it as text. The p-System gets around this problem by having several specific, predefined file types.

For the most part, you'll be working with the following file types.

File Type	Contents
.TEXT	Normal, readable text material. Generally this information is used by the Editor.
.CODE	Executable instructions, or "code," for the computer to use. The .CODE file type is used for both p-Code and machine code in the p-System. The programs you receive from Osborne and other vendors normally come to you as .CODE-type files, ready to run.
.BAD	As you can probably guess, the file type of .BAD indicates that the information in it is not usable, or of no interest. In the p-System, the file type of .BAD specifically refers to any file that occupies an area of the diskette that the computer cannot access correctly (as in the case of a damaged diskette).
.WRK	A file you are currently working on. This will most often be a text file you are editing with the Editor. Unlike the other file types, .WRK appears in the middle of this file name.

Other file types exist in the p-System, but they are primarily of interest only to users of the entire Development System, so we won't attempt to deal with them here.

Files without one of these file types are assumed to be data files. These data files are usually reserved for information used by, or created by a .CODE file.

You need not always identify the type of file you are working with. For instance, the Editor assumes that you want to work with a .TEXT file, while the eXecute option assumes that the program name you type has the file type .CODE. If you add the file type .TEXT or .CODE when using the Editor or eXecute option, your file name will not be recognized.

When you do need to identify the file type, type the entire file name, including a period, and the file type.

A special work file is used with the Editor. If you are editing a text file, and save it without a file name, it can be stored in a file named SYSTEM.WRK.TEXT. However, this Editor work file is rarely used outside of program development.

Volumes

We have mentioned that "volumes" are the devices which send, receive, and store files.

The p-System predefines several volumes.

Volume Number	Volume Name	Description of the Device
#1:	CONSOLE:	The screen and keyboard, typed characters are displayed.
#2:	SYSTEM:	The screen and keyboard, typed characters are not displayed.
#4:	disk name	The diskette in the main disk drive (usually drive A), referred to as the "system diskette."

Volume Number	Volume Name	Description of the Device
#5:	disk name	The diskette in the second disk drive, (usually drive B), referred to as the "alternate diskette."
#6:	PRINTER:	The printer, if available.
#7:	REMIN:	A serial input line, as from a modem.
#8:	REMOUT:	A serial output line, as to a modem.

In the table above, notice the conventions that are used in naming the devices. First, you may refer to any device by its number. You do so by typing:

a number sign, followed by
the device number, followed by
a colon.

Valid examples of this are:

#1:
#5:

You can also refer to volumes by their name. You do so by typing:

the device name, followed by
a colon

Valid examples of this are:

CONSOLE:
SYSTEM:
PRINTER:
REMOUT:

Another thing that might have caught your attention about the table: the two diskettes in #4 and #5 do not have predefined *disk names*. The disk name refers to the name of the diskette, and not to the disk drive itself. Every p-System diskette is given a name, which is stored on that diskette. It is this name that is used for the disk devices, not the words "disk name." For example, the Osborne version of the p-System is supplied with the following diskette.

OSBORNE:

We haven't yet told you why you use the volume identifier. (The term "volume identifier" refers to either a volume number or a volume name.) Actually, it's quite simple: the volume identifier tells the system where to get a file from or where to put it. If you want to use a file named **MYFILE.TEXT** on the first system diskette, and the volume name of this diskette is **OSBORNE**, you identify the file as:

OSBORNE:MYFILE.TEXT

You should now see why a colon can't appear in a file name; the computer would interpret the colon as being the marker between the volume to use and the file name to find.

There are two shorthand methods of referring to volumes: you can tell the p-System to use the system disk, or you can tell it to use the default (or prefixed) disk.

The system disk is the disk you used to start the p-System. If you started the p-System by putting the diskette in drive A and pressing RETURN, then the diskette in drive A is the system disk. You can refer to this system disk by typing **[*]** instead of a volume identifier. For example, if you want to use the file **MYFILE.TEXT** on the system disk and your system disk has the name **OSBORNE**, you could refer to this file by typing any of the following:

OSBORNE:MYFILE.TEXT
#4:MYFILE.TEXT
***:MYFILE.TEXT**

The second shorthand method of referring to the volume identifier is the default (or prefixed) volume. This default volume is the volume that is assumed when you don't indicate a volume. For example, if you simply type **MYFILE.TEXT** without indicating a volume, the p-System will search the default volume for that file.

When you first start the p-System, this default volume is the same as the system disk. If you want to change the default volume to something other than the system disk, use the Filer's P(refix command (described later).

When indicating a file on the default volume, just type the file name. If you want to indicate the entire volume, type and press .

Putting it All Together

We've already presented the idea of volume, file name, and file type representing the entire specification of the information you want to use while in the p-System. But to make sure that you understand it completely, let's recap:

- The complete specification for a file is the volume it is to be sent to or from, the file name, and the file type.
- Under some circumstances, you may omit pieces of information. If you omit the volume, a default device is used. If you omit the file type, a default file type is assumed. If you omit the entire file name specification, a work file is used.

Congratulations on hanging in there through all this seemingly theoretical information. In the next lesson we'll prove to you that it was all worthwhile and not theoretical at all.

File Handling

In this lesson you'll learn how to work with your files — how to create, change, and remove them. You'll learn to list their names, and to copy them. And, you'll learn a little more about how to refer to them.

The Filer

You use the **Filer** to manipulate files. You already have a little experience with it. You used two of its options, **Date** and **Zero**, when you formatted your diskette.

Once you've pressed **[F]**, for **Filer**, at the Command level, the **Filer** promptline appears.

```
Filer: G(et, S(ave, W(hat, N(ew, L(dir, R(em, C(hng, T(rans, D(ate, ? [D.7]
```

Here is an alphabetical list of the **Filer** options. The ones that are useful with your p-System are shown in boldface.

B (ad blocks	P (refix
C (hange	Q (uit
D (ate	R (emove
E (xtended Directory	S (ave
F (lip-swap/lock	T (ransfer
G (et	V (olumes
K (runch	W (hat
L (ist Directory	X (amine
M (ake	Z (ero
N (ew	
O (n/off line — used only with hard disk drives	

Because **Filer** is an integral part of the p-System, we'll look at it a little closer than the rest of the system. We will discuss only the options provided with this version of the p-System.

You'll be using the **Filer** frequently, so take the time to learn it well.

Flip-swap/lock puts the Filer into the memory of the Executive, allowing you to use the Filer even if you remove the disk.

Leaving the Filer

The **Filer's Quit** option is easy to explain: it takes you out of the Filer and returns you to the p-System command level. Need we say more?

Don't press **Q** now. We just want to make sure you know how to do it in case you have to stop in the middle of the lesson.

Creating an Empty File

There are several ways to create a file. The Editor creates text files. The Compiler (not included with this version of the p-System) creates code files. The Filer's **M**(ake command can also create a file. A file created by Make contains the information that previously existed on that disk location. You might want to use Make to recover a file that you accidentally Removed or for other special purposes.

We will now create several empty files with the Make option. Although creating empty files is not something you will normally want to do, we will be using them in this tutorial to teach you about the other options of the Filer.

Press **M**, for **Make**. You are asked what file to make:

Make what file?

Enter a valid p-System file name (remember, you can also enter the volume name and the file type, if desired).

Immediately following the file name, you can enter the number of blocks you want the file to occupy, enclosed in square brackets.

Let's make an empty file called NEWFILE.TEXT on your system diskette by responding to the **Make?** prompt

Make what file?

with:

OSBORNE:NEWFILE.TEXT[4]
and pressing RETURN

The first four contiguous unused blocks on your diskette will be reserved for the file NEWFILE.TEXT. (You could have referred to the system diskette as #4 or *.)

If you do not specify the number of blocks to reserve, the p-System looks for the largest set of contiguous blocks and assigns that to the file. Alternatively, you can assign half of the largest area of contiguous blocks, or all of the second largest area (whichever is larger), to a file by enclosing an asterisk in the brackets. For example, typing

OSBORNE:NEWFILE.TEXT[*]

will assign 10 blocks to NEWFILE.TEXT if the largest free space on volume OSBORNE is 20 blocks and the second largest area is less than 10 blocks.

If you have not already done so, make the 4-block file on your OSBORNE diskette as described above. Then make four files on the diskette in volume #5. Name them NEWFILE1.TEXT, NEWFILE2.TEXT, NEWFILE3.TEXT, and (you guessed it) NEWFILE4.TEXT. Assign each file 4 blocks. When referring to the diskette, try using both your volume name designation,

which is the name of the diskette, and the device volume name, which is #5. The file designation interaction will look something like this:

Make what file?

M Y D I S K : N E W F I L E 1 . T E X T [4]

Make what file?

M Y D I S K : N E W F I L E 2 . T E X T [4]

Make what file?

5 : N E W F I L E 3 . T E X T [4]

Make what file?

5 : N E W F I L E 4 . T E X T [4]

Wild Card Conventions

Wild cards are a shorthand way of referring to file names.

In the p-System, three wild card specifiers are used. Their meanings are slightly different from their CP/M Plus counterparts. The wild card characters are:

- = Replaces a group of characters without querying the user.
- ? Replaces a group of characters and queries the user.
- \$ Means the same file name as just specified.

In a sense, all three of these p-System wild cards function similarly to the CP/M Plus asterisk—they replace a group of characters. Only one “group” wild card can appear in a file name, although they can appear in any character position. The following are all valid uses of wild cards:

A=Z Matches all file names that start with the letter A and end with the letter Z.
= Matches all file names.
=.TEXT Matches all file names that end in “.TEXT”.
UANDIARE=.TO Matches all file names that start with “UANDIARE” and have a file type of “TO”.

If you use the ? instead of the equal sign, the action to be performed with that file will not be done until you have verified it by pressing the Y key in response to a message that appears on the screen. The action isn’t performed if you press the N key.

Looking at the Directory

The L(dir (List Directory) option lists the files on a specified volume.

Press while at the **File** promptline. The system asks you which volume you wish a listing of files for by prompting:

Dir listing of what volume?

To see a listing for all the files on a certain volume, type the volume number, its alternative designation, or the volume name (you could type #5 or *, or OSBORNE:, or : for example), and press RETURN.

Type and press .

After you press RETURN, the directory appears on the screen. The first column contains the file name. The second column contains the size of the file in blocks. The third column indicates the date each file was last updated. The summary line (on the bottom) indicates the number of files listed, the total number of files on the disk, the total number of blocks used, the total number of blocks still available for use, and the largest contiguous unused area.

To find out if the specific file NEWFILE3.TEXT is on volume #5, press and type #5:NEWFILE3TEXT and press RETURN.

To find out about a certain group of files, you can type, for example, #5:NEW?.TEXT for a listing of all file names that start with NEW and that have a file type of TEXT on volume #5. Because you used ? instead of =, the system asks you to verify the listing of each file. For example

List NEWFILE3.TEXT?

Respond with Y to list each file that begins with NEW. If you do not want to be queried for each file, use the equal sign.

If you want to save a listing in a file, specify a file name following a comma after typing the name of the volume and/or files.

Dir listing of?

#5:=.TEXT,#5:LIST.TEXT

All files that have a file type of TEXT will be placed in a file named LIST.TEXT on volume #5.

To send a list of files whose names start with NEW to the printer, type:

```
#5 NEW TEXT PRINTER:
```

Don't send files to the printer yet. We'll soon show you how to display the contents of the files you placed on #5.

Displaying the Extended Directory

The E(xt-dir (Extended Directory) option works exactly like the **List Directory** option. Use the **List Directory** option if all you want is a list of the files on a volume; use **Extended Dir** if you want more information about a file.

Press **[E]** while at the **Filer** promptline. Then answer the query as you did for **List Dir**.

The resulting display shows the:

- file name,
- the number of blocks it occupies,
- its creation date,
- its starting block.
- the number of bytes in the last block of the file, and
- its file type

Right now, experiment using wild card designations with your **List Directory** and **Extended Directory** options.

Changing File Names

You can rename files or volumes using the **Filer's Change** option.

Press **[C]**, for **Change**, while the **Filer:** promptline is displayed. To change the name of NEWFILE.TEXT on your OSBORNE diskette to RENAMED.TEXT, respond as shown. You are asked:

Change what file ?

Type ***:NEWFILE.TEXT** and press **[RETURN]**. Next, the system displays:

Change to what?

Respond with **RENAMED.TEXT** and press **[RETURN]**.

You do not have to specify the volume in response to the second question. You may use a wild card specifier. We suggest that you use the query wild card (?) if you want to change the names of a group of files.

If you wish, you can change the name of a diskette (the disk name of the diskette in volume #4 or #5) by typing only its name followed by a colon (i.e., not typing any file name after it).

Removing a File From a Diskette

To erase a file from a directory of a volume, you use the **Filer's Remove** option.

When you press R, for **Remove**, the system answers with:

Remove what file?

Type M Y D I S K : N E W F I L E 4 . T E X T
and press RETURN to remove your file NEWFILE4.TEXT from
your diskette.

You can erase groups of files using a wild card specifier.

After the erasure, the system asks:

Update directory?

Answer Y to delete the file from the directory. **N** cancels the
removal request.

Now use the L option to check the directory.

Listing Your Volume Names

To find out what volumes are currently on-line, you use the **Filer's Vols** option. A disk is on-line when it is a proper p-System diskette and is correctly placed in a drive. Other volumes, such as PRINTER:, are on-line when they are properly connected and turned on. If a volume is on-line, a number sign (#) appears next to it. This one's easy. Just press the V key, and the list of your volumes appears — that's all there is to it. It also lists the number of blocks on your diskette, the system disk (called the root volume), and the current prefixed (or default) volume.

The **Vols** option is handy for figuring out which disks are in your drives, whether your printer is properly connected, and so forth.

Press V to see the names of all your volumes.

Copying Files

The **Filer's Transfer** option transfers a file from one location to another. It may be from one volume to another, or from one location to another location on the same volume. The **Transfer** option does not pick up the file and move it; it *copies* it. Therefore, you end up with two copies of the file in question. p-System's **Transfer** is analogous to the CP/M Plus **PIP** command.

First, press **T** to get the **Transfer** option. From there, you can perform several types of transfers.

Pressing **T**, for Transfer, results in the following message:

Transfer what file?

After you enter a valid p-System file name, the system prompts you with:

To where?

Many possibilities are available to you using the Transfer command; so we will summarize them for you.

Copying a Single File

In response to the first prompt, specify the volume and file name of the file to be moved, then the volume and file name for the destination. If the destination file name is to be the same as the source file name (i.e., the same file name on two different diskettes), you may abbreviate the file name in the second response to a dollar sign (\$). For example, to transfer the file NEWFILE1.TEXT on MYDISK to the OSBORNE diskette, type:

```
Transfer what file? MYDISK:NEWFILE1.TEXT
```

```
To where? *$
```

You have just copied the file NEWFILE1.TEXT to the OSBORNE diskette. The dollar sign \$ gave the copy the same name as the original file.

To Copy a File on the Same Drive

Respond as described below to the first prompt, then use the same volume name and a different file name in response to the second prompt. You may also include square brackets to indicate the number of blocks for the new file. The new file will be placed in the first area on the disk that is at least that many blocks long.

Let's make another copy of your file NEWFILE1.TEXT on your OSBORNE diskette and name it COPYFILE.TEXT.

```
Transfer what file? #4:NEWFILE1.TEXT
```

```
To where? #4:COPYFILE.TEXT
```

If you copy to the same diskette, remember to give the file a different name from the original. You may also insert another disk to copy to, when prompted.

To Copy Several Files

Use the wild card character in part of the file name you specify to the first prompt; use either the dollar sign (\$) shorthand indicator or another wild card-embedded file name in response to the second prompt.

To transfer all the files that end with .TEXT from the disk in #4 to the disk in #5, do this:

Transfer what file? #4?.TEXT

To where? #5:\$

You are queried for each file to be transferred. If you answer Y to each file, all .TEXT files are copied from #4 onto #5.

Press Y to copy each file.

Because some of these files have the same name of both disks, you may have to remove some of the files on volume #5 when prompted by Transfer.

To Copy an Entire Diskette

First format a double-density diskette using the CP/M Plus COPY program, or the p-System Format option.

Press T at the **Filer:** promptline. The system responds:

Transfer what file?

Remove the system diskette and replace it with the empty formatted one. Place the disk to be copied from in #5.

Now, type # 5 and press RETURN to indicate the source drive. When the systems asks:

To where?

type `#4` and press `RETURN` to indicate the destination drive. Now the system asks:

Transfer 390 blocks: (Y/N)

Answer `Y` if you want the entire diskette copied.

You can hear the disk drives work. When the copying stops, put the system diskette back into its drive. Always return the system diskette to its drive before you press another key, or you will have to restart the system. You can create a new OSBORNE diskette in the same way, except that the source drive is #4 and you do not remove the OSBORNE diskette. Put the blank diskette in #5. You can also use the **Back-up** option from the **UTIL:** promptline to perform this function. It is described in the section *More About Diskette Preparation and Handling*.

To Copy to the Screen or Printer

The Transfer option also copies files to the screen and the printer. Remember that you have a file named LIST.TEXT on MYDISK. To see the contents on the screen, respond to the query:

Transfer what file?

with `#5:LIST.TEXT`. In response to:

To where?

type `#1:` or `CONSOLE:` and press `RETURN`. This will display a file's contents on the screen.

To print a file, respond to:

To where?

with **#** **6** or **P** **R** **I** **N** **T** **E** **R**: and press **RETURN**. This will print the file on a printer hooked up to your Executive. Make sure that your system diskette is properly configured for your printer with the **UTIL's Configuration** option.

An Easy Way to Refer to Volumes

We've mentioned before that you can use the asterisk to substitute for the current, system disk. When you first start the p-System, this system disk is also your default volume. You can use the **Prefix** option to change the default volume.

Prefix does two things: it provides you with a shorthand way to refer to a volume and it tells the computer which volume to search first for a file.

Here's how it works. Press **P** when the **Filer** promptline is on the screen. This prompt appears:

Prefix titles by what vol ?

You can respond to it in a variety of ways:

If you type **#** **5**, you don't have to include the volume number when referring to a file on that volume. However, if the proper response to a prompt is the volume number only, you will have to type the colon at least. Otherwise, you will be sent back to the promptline. In addition, the p-System will

automatically look on that volume for files instead of on the system disk. You can still refer to volume #4 as * as long as it contains the system disk.

You can also respond to this prompt by typing a **disk name** such as **[M][Y][D][I][S][K]**. Now the system will look for a file on a diskette named MYDISK in either disk drive.

We suggest that, after changing the date, you change the prefix volume to "#5:" (i.e., the second disk drive) before starting to use the system each day. Then, if you want to access something in drive A, you type **[*]** before the file name; if you want to access something in drive B, you simply type the file name. If only the volume number is required, then you will type **[.]**.

Work Files

Work files are files you are in the midst of working on. They are almost always used for program development. Since you can't develop programs without the full Development System, there is no reason for you to know a lot about work files.

However, since there is a chance you might accidentally create a work file with the Editor, we will briefly explain the Filer options for dealing with work files: **S**(ave, **G**(et, **W**(hat, and **N**(ew).

Creating a work file

There are two ways to create a work file: using the **G**(et option of the Filer, and using **U**(pdate to exit from the Editor.

Get designates an existing file as a work file. For example, if you had the file **TOSAY.TEXT** on your disk and you wanted to make it a work file, you would press **G** for Get from the Filer promptline and then type **TOSAY.TEXT** and press RETURN. This will make **TOSAY.TEXT** a work file.

The **U**(pdate option of the Editor also creates a work file. If you use update to save a text file, the Editor will save your text in a file called SYSTEM.WRK.TEXT. (We will describe the Editor later.)

In summary, there are two ways to create a work file: designating an existing file as a work file with the Get option of the Filer, and creating a special work file called SYSTEM.WRK.TEXT with the Update option of the Editor.

What work files do

Once you have created a work file, certain Command options will automatically assume that you want to work with that file. The only important one to know about is the Editor.

If you have a work file, pressing E for Editor will automatically load that file into the Editor. You cannot edit another file as long as you have a work file available.

If you were developing programs, you might find this feature convenient. However, mostly when you are editing text you will find it annoying.

Finding out what your work file is

If you want to find out what your work file is, press **W** for **W**(hat from the Filer promptline. This will tell you your current work file, if you have created one with Get or Update.

Getting rid of the work file

If you want to be able to choose which text file to edit, you must first get rid of the work file. There are options in the Filer which allow you to do this: **S**(ave and **N**(ew).

S(ave saves a file called SYSTEM.WRK.TEXT (this is the file created by using Update in the Editor) under another name. You can either replace your original text file with the updated

version in SYSTEM.WRK.TEXT, or save the updated text with a new name. When SYSTEM.WRK.TEXT has been copied to another file, the file SYSTEM.WRK.TEXT is automatically erased.

N(ew works two ways, depending on what kind of work file you have. If your work file is SYSTEM.WRK.TEXT, it will erase this file without saving it. If you designated an existing file as your work file with Get, it will undesignate it, but won't erase it.

In summary, if you want to stop using SYSTEM.WRK.TEXT as your work file, choose **S**(ave to save it first, or **N**(ew to erase it without saving it.

If you want to stop using any other text file as your work file, choose **N**(ew.

You can perform the basic file handling chores now. In the next lesson you will learn how handle problems with diskettes and files.

Handling Diskette Problems

This lesson shows you how to get out of trouble when you have physical problems with a diskette or when you accidentally erase a file or directory that you really want to keep.

Finding Bad Blocks

Every once in a while you may run into a diskette that has something physically wrong with it. The p-System provides you with a way to find out where these bad areas are, and then attempt to fix them or at least mark them so that the computer won't try to write on them.

Use the **Filer's Bad Blocks** option to check the diskette for blocks that are faulty. Press **[B]** when the Filer promptline is on your screen.

First, the system asks you the name of the diskette to scan. To check the diskette named MYDISK (or whatever you named it), you use this interaction:

Bad block scan of what vol?

Type **[M][Y][D][I][S][K][:]**

Now you are asked:

Scan for 390 blocks? (Y/N)

Answer Y. This means that there are 390 blocks on the diskette to be scanned, and the p-System wants to verify that you want all of them scanned. Scanning begins. If you had replied N, you would have been asked:

Scan for how many blocks?

You would have replied with the number of blocks you wanted scanned. You can use this second query if you want to look for bad blocks in a particular area. (For example, you may be having trouble with a file.) If you ask to scan 50 blocks, the scanning starts at 390 and goes backward to 340; you can also specify a range to scan.

Examining the Bad Block Area

You can use the **Filer's Examine** option to attempt to correct a problem in a file indicated by the **Bad Blocks** option.

Examine attempts to recover the data on the file, and mark the bad blocks that can't be recovered so that they cannot be used again.

To execute the **Examine** option, press X, for **eXamine** when the **Filer** promptline is on the screen. The following message appears:

Examine blocks on what vol?

Specify a valid p-System volume name or number, or use one of the shorthand methods of specifying a volume (* or :). Next, the computer displays:

Block-range?

You enter a range by typing the first block to examine, a hyphen, then the block number to end the examination with. You might enter:

Block-range? 1-10

and press `RETURN` for instance. You could also enter a single number to examine a single block.

The p-System tells you the files in those blocks that might be endangered, and then asks you:

Fix them?

For now, *make sure you answer N*. You could answer either **Y** or **N**. If you answer **Y**, the option attempts to save the rest of the file and mark what it can't save as a bad block.

Examine cannot always fix a bad block. In that case, the prompt:

Mark bad blocks?

is displayed. Be careful about hastily replying with **Y** to this question. Once a block has been marked as bad by the p-System, the information that is in that block is totally lost.

Rearranging Files on the Diskette

The **Filer's Krunch** option consolidates ("crunches") the unused areas on a diskette to make the best use of them.

After you've been using a diskette for a while, the changes you've made to the files may result in several small, unused spaces. For instance, if you've erased a file, the space formerly used by the file is unused. To remedy this situation, you use **Krunch**.

To crunch the data on your diskette MYDISK, here's the interaction:

Press **K**, for **Krunch**, then:

Crunch what vol?

You type the volume name (or number) of the volume you want to have consolidated. Type **M Y D I S K .** or **# 5 .** or **.** if you set the prefix. Then press **RETURN**.

Next, the system asks if you want the entire diskette crunched.

From end of disk, block 390? (Y/N)

If you type **Y**, files are moved toward the beginning of the diskette so that all the unused space is at the end. Normally, you would reply with a **Y** to this question. If you want the unused area to be somewhere else on the diskette, you would answer **N**, and the system would ask where you wanted the crunching to start. You will normally answer **Y** to this question.

Press Y.

Never use krunch on a diskette with bad blocks which have not been marked with eXamine.

Recovering Erased Files

Use the **Recover** program to recover files that have been accidentally erased. When files are removed, they are still there, but the system can't find them because the directory entry is gone. **Recover** finds lost files and creates an entry for them in the directory.

Right now, use the **Filer's Remove** option to erase the file LIST.TEXT on volume #5. Now we can show you how **Recover** works.

We've spent a lot of time in the **Filer**. To demonstrate Recover, however, we have to go back to the **Command:** promptline and press X to execute the RECOVER program. After you have pressed Q to Quit the filer and X to eXecute a program, the following prompt appears on your screen:

Execute what file?

Respond with Recover or *Recover to start the program running. The program asks:

USER'S DISK IN DRIVE# (0 exits):

For our sample case, type 5, and press RETURN.

Next, the system wants to know the the name of the volume:

USER'S VOLUME ID:

Type M Y D I S K and press RETURN. This does not have to be the name you originally gave the volume. This name will be given to the volume when Recover is finished, however. Now **Recover** displays the files it has found in the current directory and asks:

Are there still IMPORTANT files missing (Y/N)?

Press N if it displays all the files you want; press Y if there are other files it has not found. **Recover** then searches areas not accounted for by the current directory.

To recover LIST.TEXT, press Y.

Since **Recover** cannot determine the original name of a text file, it labels it DUMMY###.TEXT, with ### standing for unique numbers. If the extended directory shows that there are an odd number of blocks in a text file, the end of the file was probably lost. Code files are given the name which corresponds to the program name. This may not have been the original code file name, however. Data files cannot be recovered.

When **Recover** has finished its search, it asks if you want to update the directory.

GO AHEAD AND UPDATE DIRECTORY (Y/N)?

Type Y to update the directory, N to abort the operation. If you press N, you will be unable to access the recovered files

Another Way to Recover Lost Files

If you accidentally use **Remove** to erase a file from the directory of a diskette, sometimes you can recover it by making a new file with **Make**. First use the **Extended Directory** option to find where the file you removed might be. (It will be in an unused portion of the diskette.) Now, use **Make** to create a new file the same size as the erased file. **Make** automatically puts the new file in the first available space which is at least the size you specified, so you might have to create some dummy files to use up unused areas near the beginning of the disk. If this works, the new file you have created in the old file's place will contain the old file's data.

Preventing Directory Trouble

So far in this lesson we have dealt with how to recover files. You can recover an entire directory and prevent trouble by using two utilities that create and copy a duplicate directory on your diskette — **Markdupdir** and **Copydupdir**.

Markdupdir marks a location on diskette for a duplicate directory. Whenever an entry is made in the regular directory that you created with the **Zero** command, an entry is made in the duplicate directory as well.

Before starting **Markdupdir**, check to see that blocks 6 to 9 are free by using the **Filer's Extended Directory** option. If they are not free, you may want to use **Transfer** to transfer the files in those blocks to other free blocks on the diskette.

At the **Command:** promptline, press to execute a program, then tell the system that you want to execute the program

contents of the primary directory to be overwritten with the contents of the duplicate directory.

If all goes well, you see this message:

Directory copy is complete. Type <ret> to exit.

As you can see, **Markdupdir** and **Copydupdir** may save you from going through some of the more complex processes if you ever need to recover an entire directory.

Some Useful Commands

This section describes some useful functions.

Reinitializing the System

The **Initialize** command is used to restart the system, or to place it in its initial state, as if you had just rebooted (reloaded) the system. When the **Initialize** command is used, most of the p-System's internal options, variables, defaults, and so on, are set back to their original state.

To use the **Initialize** command, press when the **Command:** promptline is on the screen.

Your screen will go blank, the drives will whirl, and the Command promptline will reappear on your screen.

```
Command: E(dit, R(un, F(ile, C(omp, L(ink, X(ecute, A(sssem, D(ebug, ? [IV.12 B]
System re-initialized.
```

Rerunning the Last Program

User Restart executes the last program you ran. For example, if you have just run the **Filer**, User Restart will cause the Filer to be rerun.

To use the **User Restart** program, press when the **Command:** promptline is on the screen.

Leaving p-System

The **Halt** command stops the p-System immediately. All memory is erased from the computer.

As you can see, **Halt** is a drastic action, because the only way you can get back into the p-System is to press the RESET button and press **RETURN**.

To use **Halt**, press **[H]** when the **Command:** promptline is on the screen.

Saving Your Commands in a File

You can cause the p-System to remember a sequence of commands you give it, and to save those commands in an executable text file you name. The **Monitor** option performs this task for you.

The monitor allows you to “program” an oft-repeated sequence of keystrokes, instead of actually having to type them every time.

To use the **Monitor**, press **[M]** when the **Command:** promptline is on the screen. The **Monitor** displays this promptline:

Monitor: B(egin, E(nd, A(bort, S(uspend, R(esume

Here is a description of each option:

- B**(egin Opens a file in which the monitored commands are to be kept. Monitoring does not begin until you press **R** for Resume.
- E**(nd Completes the monitoring process and closes the file that was saving your input. You must End a monitoring session in order to have a usable input file.
- A**(bort Cancels the monitoring process without saving anything into the file you specified.
- S**(uspend Returns you to the command level without closing the file you specified with the Begin option. Can be used during a monitoring session to suspend the monitoring process temporarily.
- R**(esume Starts the monitoring session and returns you to the command promptline. Also restarts the monitoring process after it has been suspended.

Press **B** to start the monitoring sequence. The **Monitor** asks you:

Write to what file:

Type, for example, **# 5 : C O M M A N D . T E X T** and press **RETURN**.

Press **R** , for **Resume** to return to the Command level and begin monitoring.

Now give a couple of commands like **F** to get into the **Filer** and **L** to see the directory of your MYDISK diskette. Get out of the **Filer** by pressing **Q**.

From the **Command:** promptline, press **M** to return to the **Monitor** and then press **E** to end the monitoring session.

Press **R** or **S** to return to the Command propmtline.

To execute all of the commands in the file, press **X** at the **Command:** promptline to execute a file. The system asks you for the file name to execute. You respond:

```
I=M YDISK:COMMAND.TEXT
```

Everything you did between entering and leaving the Monitor is repeated.

More About Diskette Preparation and Handling

This section explains how to put "boot tracks" on a diskette. Boot tracks allow a diskette to be used in the main disk drive at start-up time. You will also learn how to use the p-System to back up a diskette.

All of these facilities reside in the UTIL.CODE program.

First make sure the **Command:** promptline is on the screen. Then press for **eXecute**. You respond to the prompt

Execute what file ?

by typing and pressing .

The system displays this promptline.

UTIL: F(ormat B(ack-up C(onfig M(ake boot Q(uit

You have used the **F** option of the UTIL program to format a diskette, so there is nothing new about responding to this promptline.

Pressing **ESC** at any time will return you to the UTIL promptline.

To return to the **Command:** promptline, press **Q**.

Putting Boot Tracks on a Diskette

The **Make Boot** option places boot tracks (or system tracks) in the first 12 sectors of a formatted diskette.

First, press **M** to access the **Make Boot** option. The system responds with:

```
Bootstrap Copy Utility
```

```
Enter bootstrap source disk unit number (4,5)
```

Respond with **4** if that is where the system disk with bootstrap tracks is located. If you choose to have the source diskette in drive B, type 5. If you change your mind about the whole thing, press RETURN to exit. Follow your entry with **RETURN**. Now the screen says:

```
Place SOURCE disk in unit #4.  
Place DESTINATION disk in unit #5  
and press RETURN when ready.
```

You can remove the system diskette and put another diskette in its place. You might do this if another diskette has the boot tracks configured just the way you want them. Just make sure that the diskette you are using for your source actually has boot tracks on it.

Before pressing RETURN, place the source diskette in the volume you indicated, the destination diskette in the other volume. Now you can press **RETURN**.

When M(ake boot is complete, put the system diskette back in drive A if you removed it.

Configuring Your Boot Tracks

Use the **Configuration** option to set up the boot tracks for your system. The p-System's configuration option is similar to the CP/M Plus Setup program.

Your new diskette is in drive B.

First, press from the **UTIL:** promptline. The system then asks for the source of a configuration:

Enter source disk unit number (4,5)
or (1) for memory:

We'll get our configuration parameter from the OSBORNE diskette, so type 4 and press . You could also press 1 if you have stored the configuration in memory. The configuration parameters in memory were read from the system disk when you first started the p-System. If you decide not to configure the diskette, press RETURN to exit without reading any configuration parameters.

This menu now appears on your screen:

```
Config: P(rinter R(emote B(aud I(nit string Q(uit  
  
PRINTER: Centronics Parallel  
REMOTE: Standard Serial  
Baud Rate: 1200  
Printer Initialization String  
(None specified)
```

Press either **P**, **R**, **B**, or **I** to make changes to the configuration. Each time you press one of these letters the screen entry changes. Stop pressing the letter when you get to the configuration you want.

Here's how the changes work:

PRINTER: **P** changes the printer setting. Each time you press **P**, a different setting is displayed. Check your printer manual for the correct setting.

REMOTE: **R** changes the remote port from standard serial to disabled.

BAUD RATE: **B** changes the baud rate from 1200 to 300. Mostly you will want to have this set at 1200.

PRINTER INITIALIZATION STRING: This information is fairly technical, but it may be necessary to use it.

Press **I** to create a string of bytes which will be sent to your printer to initialize it the way you want it. This string is sent to the printer each time you access it. When you press **I**, this line is displayed on the top of the screen:

(0-9, A-F, <bs>, <return>, <esc>)

You may enter as many bytes as you like by typing the hexadecimal digits (0-F). To correct mistakes, use the left-arrow to back space and erase. Press RETURN when you are satisfied with the string. Press ESC to exit without changing the current printer initialization string.

If you select an IEEE printer, you will see an additional prompt on your screen. This is the Address Port. Press **A** to change the address port for your IEEE printer. When you press **A**, this prompt is displayed

Address (<0 . .>, <esc>)

You should enter one or two digits in the range 1 to 30. After you have entered the desired port address, press **ESC**.

When you have finished with configuring your disk, press **Q** to Quit. Your screen says:

Update to memory(1) or disk (4,5):

Answer **5** and press **RETURN**. If you were going to configure several diskettes you might have responded with 1 so that you could get the same configuration from memory for each diskette. You might also use 1 if you want to test the new configuration parameters before saving them on your disk.

If you don't want to update to disk or memory, press **RETURN** instead of typing a number.

Press **RETURN** when you have finished your changes.

Backing Up Your Diskettes

UTIL's **Back-up** option creates a sector-by-sector copy of a diskette onto a blank diskette. You can copy onto a diskette that already contains data, of course, so think twice before you do it. **Back-up** will copy both an Osborne double-density diskette and a Universal Medium diskette (described in the next lesson).

From the **UTIL:** promptline, press **B**

```
Disk Copy Utility
Enter source disk unit number (4,5)
```

Type **4** and press **RETURN**. You could also have typed **5**, or just **RETURN** to exit from the option.

```
Place SOURCE disk in unit #4
Place DESTINATION disk in unit #5
ANY DATA IN DRIVE 5 WILL BE ERASED.
and press RETURN when ready
```

Place the source diskette in the volume you indicated (**#4** or **#5**), the blank diskette in the other drive, and press **RETURN**. Be sure you have the correct disks in both drives.

Back-up formats the new disk and then copies the source disk onto the new disk. When it is finished, remove the diskette from drive A and replace it with the system diskette.

Type **Q** to return to the **Filer** promptline.

Just a note: you can use **Back-up** to copy the system diskette. Just leave the system diskette where it is and copy it to the diskette in the other drive.

Running Applications Programs

This lesson tells you how to get set up to use an application program. You should already be familiar with all the tasks required.

Determining Diskette Format

Okay, you've received a piece of software that uses the UCSD p-System, so how do you proceed?

First, you must determine the disk format:

- Osborne double-density format
- UCSD p-System Universal Medium diskette format

If the diskette you acquired is in any other format, you'll have to get it converted to one of the Osborne-useable formats by a friend, your dealer, or your local user group. This can be very difficult, so we suggest that you make sure you can get the program in the proper format before you purchase it. (Note: if the program was supplied to you on paper in source-code form, you might want to purchase the UCSD Development System from the Osborne Approved Software program; use the Editor to enter the program, and then the Compiler to make it usable.)

You should also be sure that the developer of the application program has not written any machine-specific code which will not run on the Osborne Executive. For example, if it includes assembly language routines designed for another computer, it may not run on the Osborne Executive even though the p-System is used in both cases.

As a safety check, it is a good idea to ask the developer if the program has been tested on the Osborne Executive.

Making an Exact Copy of Master Diskette

To make an exact copy of an application program diskette, use the UTIL program (described earlier). From the UTIL promptline, first choose the **B(ack up)** option to make a copy of the original diskette. Put this master diskette in a safe place.

To run this program, put your copy of the application program diskette in volume #5, your copy of the system disk (which you received with your Executive) in volume #4, and press X to eXecute from the Command promptline. When asked for the program to execute, type # 5 : and the name of the program.

Follow these steps first before you use your new program. If you want to alter the copy in any way, follow the instructions below after you have made an exact copy.

Altering the Copy

You may want to alter your copy of the application program for two reasons: to place it on a system diskette, or to place it on a diskette with more room.

Placing your program on a system diskette

If you place your application program on a system diskette, you will be able to store the system files and your program files in volume #4, and keep volume #5 for any data files your program creates.

To use the p-System, your system diskette must have the following files:

SYSTEM.PASCAL
SYSTEM.MISCINFO
SYSTEM.INTERP
SYSTEM.CHARSET

If any of these files are missing, the p-System will not run properly when you put it in volume #4 and press RETURN.

You will probably also want SYSTEM.FILER (to use any of the Filer options) on your system diskette.

To make a diskette with these SYSTEM files and the files in your application program, first use Extended Directory to make sure that these files take up less than 390 blocks. If they take up more than that, you will have to keep your system files and the application program's files on separate diskettes.

If there is enough room, use the Format option of the UTIL program to Format a new diskette, and then use the Make Boot option to put boot tracks from your p-System diskette on this new diskette.

Next, use Zero to initialize the directory of the new diskette. You should initialize it with 390 blocks.

Finally, transfer the files SYSTEM.PASCAL, SYSTEM.MISCINFO, SYSTEM.INTERP and SYSTEM.CHARSET from your p-System diskette to the new diskette. Then transfer the application program files to the new diskette.

Making a copy with more blocks

Osborne, IBM, SofTech Microsystems (the originator of the p-System), and several other companies all use a common "Universal Medium" diskette format for transferring programs between machines.

This format is basically the IBM single-density diskette format (approximately 156K of data storage capacity on a single diskette). Although you can use programs on diskettes in this format with any Executive computer (the system automatically recognizes that such a diskette has been inserted), we recommend that you use Osborne double-density format. The reason for this is simply that the Osborne format provides about 20K more diskette storage capacity (about 12 percent more) than the Universal standard format. Most programs that use the p-System make good use of the extra 40K (two diskettes) the Osborne format provides.

To transfer an application program in Universal Medium format to a diskette in Osborne format, first use the Format option of UTIL to format a new diskette, then use the M(ake boot option to put the boot from your p-System diskette on this new diskette.

Next, use the Filer's Z(ero option to initialize this new diskette with 390 blocks.

Finally, use T(ransfer to transfer the files from the application program diskette to your new diskette.

You may also want to make this a system diskette by following the instructions in the section above.

One Final Note on Running Application Programs

The p-System you received has an interpreter (called SYSTEM.INTERP) and an operating system (called SYSTEM.PASCAL) which do *not* perform floating point arithmetic. This system should be able to run the majority of application programs.

Some application programs, however, require floating point arithmetic. If you try to use a program that requires floating point arithmetic with your system, you will receive error messages, such as:

**EXECUTION ERROR #17 - Real Size Incompatible Type
<space> to continue**

or

EXECUTION ERROR #11 - Unimplemented Instructions

These errors tell you that your system is not properly configured for the application program you are trying to use.

To run these application programs, we have provided two other interpreters on your disk. We have also included units to be placed in your SYSTEM.PASCAL file. These files are:

4WORD.INTERP and REAL4.CODE
(for programs requiring 4-word (64 bit) real number support)

and

2WORD.INTERP and REAL2.CODE
(for programs requiring 2-word (32 bit) real number support)

The majority of programs using floating point arithmetic will use the two-word system, but a small number of newer applications require the 4-word system. If you don't know which of these your program requires, ask your dealer or the vendor.

To create a new system disk with the appropriate files, follow the steps below.

Place a copy of the p-System diskette you received with your Executive in Volume #4.

Place a blank diskette in volume #5, then press X for eXecute. When asked what file to eXecute, type one of the following:

I=ZERO.BOOT This creates a new diskette with the same system as your original p-System diskette. (It will not be able to use floating point arithmetic). Use this only if you want to change back to the original SYSTEM files after using TWO.BOOT or FOUR.BOOT.

I=TWO.BOOT This creates a new diskette which uses two-word real numbers.

I=FOUR.BOOT This creates a new diskette which uses 4-word real numbers.

After you have typed one of the three choices and pressed RETURN, your new diskette will have the correct system on it. Use T(ransfer to copy over the appropriate files for your application.

SECTION 2
p-System Editor

Introduction

The p-System Editor allows you to create, alter, and examine text files—such as memos or manuscripts. It can also be used to write programs.

This explanation of the p-System Editor is divided into two parts: A tutorial and a reference guide. In the tutorial, we will lead you step-by-step through the more frequently-used options of the Editor, providing a sample of the Editor's features. The reference section covers each of the features in detail.

Although the Editor is easy to use, it may behave differently from other text editors you have worked with. We suggest that you follow the tutorial closely, and save experimenting with the Editor's features until you reach the reference section.

Overview

To use the Editor, press **[E]** from the Command promptline.

The p-System loads the Editor and the following prompt appears:

```
>Edit: No work file is present. File? ( <ret> for no file)
```

The promptline tells you that **"No work file is present."** We will explain work files in the reference section.

There are three things you can do from this prompt:

1. Press the key labelled ESC. This exits you from the Editor and returns you to the Command menu.

2. Press RETURN. This will create a new file.
3. Type the name of an existing text file and press RETURN.

For now, let's create a new file. Press `RETURN`.

The Edit Menu

When you press RETURN, you will see the following menu on your screen. This is the main Edit menu or promptline. Let's examine it briefly.

```
>Edit: A(djust C(opy D(el F(ind I(nsert J(ump K(ol M(argin P(age ? [IV.1 F6c]
```

The available options

Like the other promptlines in the p-System, the Edit promptline lists the available options, such as A(djust, C(opy, and D(el. To pick an option, press the first letter of its name. For example, to pick A(djust, you would press **A**.

At the end of the promptline is a question mark (?). This means that there are more options available. Press `?` and you see the remaining options.

```
>Edit: Q(uit R(plc S(et V(erify X(ch Z(ap [IV.1 F6c]
```

The direction indicator

At the beginning of the promptline is a right-angle bracket (`>`). This is called the direction indicator or the global direction. The direction indicator affects the direction of certain commands. For example, if you are deleting text and the direction indicator is pointing to the right (that is, you see a

right-angle bracket, as in this example), when you press the SPACE BAR it will delete characters to the right of the cursor.

Now, press the left-angle bracket key, \llcorner . Your Edit menu now shows < as the first character. This means that, when you press the SPACE BAR, the cursor will move to the left.

This direction indicator only affects the SPACE BAR, the RETURN key, and the TAB key, and it only works with some of the Edit commands.

We will come back to the direction indicator when we discuss cursor movement.

For now, press \lrcorner to change the direction indicator back.

Now, let's examine the rest of the screen.

The Window Into the File

When you start the Editor, you will see the main Edit promptline followed by 23 blank lines. Occasionally, part of a previous menu will remain on your screen. These lines are not actually part of your file, and they will disappear when you type over them.

The 23 lines below the Edit menu are your "window" into the file. Although you can access the whole file with Editor commands, you can view only 23 lines of it at a time through the window in the screen. When you choose an option that moves you somewhere else in the file, the window moves to show you that new portion of the file.

The Cursor

The cursor is a small rectangular box on your screen. The cursor shows you where the action you choose will take place. For example, when you are inserting text, the text will appear to the left of the character on which the cursor rests.

You can move the cursor to any location in a file. On your screen you will see the portion of the file that surrounds the cursor. To see another portion of the file, move the cursor.

In our examples, we will use an underline to show you where the cursor is. For example, in the following sentence

This is just to say

the cursor is under the 'u' in 'just'.

Editing Text

Moving the cursor and editing text are separate actions. To change a section of text, first move the cursor to the area you want to change. Then choose an option from the Edit menu. Once you choose an option, moving the cursor affects the text. The only time you can move the cursor without affecting the text is when you see the Edit menu on the screen.

Notation Conventions

Before we begin using the Editor, we will explain some of the notations used in this manual and on your screen.

<return> Press the key labelled **RETURN**.
<ret>
or RETURN

<space> Press the **SPACE BAR**.
or SPACE

<tab> Press the key labelled **TAB**.
or TAB

<esc> or ESC	Press the key labelled  .
<etx> or ^C	Hold down the key labelled  while you press C.
—	The underline indicates the position of the cursor on your screen.
^	Hold down the key labelled  while you press the indicated letter. See the discussion of ^C above. This is different from typing the actual ^ symbol (above the number 6 on your keyboard).
<arrow-keys>, <vector-keys>, or arrow-keys	Press one of the keys labelled with an arrow. These arrow keys are listed next.
<left-arrow>, left-arrow, <bs>, or <backspace>	Press the key labelled  .
<right-arrow> or right-arrow	Press the key labelled  .
<up-arrow> or up-arrow	Press the key labelled  .
<down-arrow> or down-arrow	Press the key labelled  .
— or ^Q	Hold down the key labelled  while you press Q.

There are other notations that are used with some of the Editor commands. We will discuss these when we discuss the commands in more detail.

Lesson 1: Beginning To Use the Editor

Inserting Text

You are now ready to begin using the Editor. On the screen you see the Edit menu. To begin inserting text, press **I**. The following menu appears.

```
>Insert: Text {<bs> a char, <del> a line} [<ext> accepts, <esc> escapes]
```

Your cursor is in the upper left corner of the screen. Type the following. Don't worry about making mistakes.

```
This is just RETURN  
to say RETURN  
I have RETURN  
eaten
```

As you type, you may make mistakes. To correct a mistake, use the left-arrow key. The cursor moves back one space, erasing characters as it moves. Press the left-arrow key until you erase the mistake, and then retype the characters you erased.

When you have finished your corrections, you will see the above example on your screen with the cursor in the space after the word 'eaten'. Let's erase the entire line. Press **^Q** (hold down the key labelled CTRL while you press Q). The word 'eaten' disappears. Now, retype the word 'eaten' and press **RETURN**.

When you are in I(nsert, there are only three things you can do to type and correct text:

1. Insert text by typing.
2. Use the left-arrow key to backspace and erase text, and
3. Use ^Q to erase an entire line.

None of the other arrow keys move the cursor while you are in I(nsert. If you press them, you see only question marks.

When you begin inserting more text, you may find it inconvenient to erase and retype lines of text each time you make a mistake. In the next lesson, we will show you an easier way.

For now, let's exit from I(nsert and return to the Edit menu.

On the I(nsert prompt it tells you that "<etx> accepts" and "<esc> escapes". Press ^[(this is what <etx> means) to save the text you have just inserted. If you had not wanted to save this text, you would have pressed ESC.

Moving the Cursor From the Edit Menu

You are back in the Edit menu. Let's try moving the cursor now.

As we explained earlier, when you see the Edit menu, you can move the cursor without affecting the text. This is part of the solution to the question "How do you make corrections without erasing and retyping lines of text?" When you are in Edit, you can move the cursor back to the area you want to correct without erasing anything. We will discuss making corrections in the next lesson. Now we will practice moving the cursor.

Try using the four arrow keys listed below to move the cursor around the screen.

-  Moves the cursor down
-  Moves the cursor up
-  Moves the cursor right
-  Moves the cursor left

The cursor will only move you over existing characters. For example, put your cursor over the 't' in 'to' and press . The cursor moves to the space after the 't' in 'just'. Press  and it moves you back to the 't' in 'to'. It doesn't move you to the right on the first line because there are no characters there.

Now try moving the cursor with these keys:

- | | |
|--|--|
|  | Moves the cursor one space |
|  | Moves the cursor to the next tab stop |
|  | Moves the cursor to the beginning of the next line |

When you have experimented with these seven keys, we will show you two more commands that affect cursor movement.

The direction indicator

Move your cursor to the last line of text on your screen and press . The direction indicator at the start of the Edit menu changes to <. Now try pressing the seven cursor-moving keys. This direction indicator doesn't affect the arrow keys, but RETURN, SPACE BAR and TAB now move you backwards. Now press  to change the direction indicator and we will show you another way to move the cursor.

Repeat Factors

Move your cursor to the 'T' in 'This'. Type **[2]** and press **[↵]**. You move down two lines. Type **[4]** and press the **[SPACE BAR]**. You move over four spaces. The number you type is called a **repeat factor**. It means "Repeat the following command this number of times." You can repeat any of the cursor-moving keys by typing a number and then pressing the key.

Move your cursor back to the 'T' in 'This'. Type **[/]** and press **[↵]**. Your cursor jumps to the last line of text. Type **[/]** and press **[↵]**. Your cursor moves to the 'T' in 'This'. The **/** is called an infinite repeat factor. It means "Repeat the following command until you run out of text." You can use this infinite repeat factor with any of the cursor-moving keys by typing **/** and then pressing the key.

Leaving the Editor

The text you have typed so far is stored in something called the "workspace." This is an area in the memory of your computer. This workspace is erased when you return to the Command promptline. If you want to save the changes you have made, you must save them in a file on your disk. Once you have saved them to a file, you can return to the Command promptline without losing your changes.

To save your text on disk, press **[Q]** to **Q**uit. You will see the following:

```
>Quit:
U(pdate the work file and leave
E(xit without updating
R(eturn to the editor without updating
W(rite to a file name and return
```

Your options are:

U, for U(pdate, saves your work in a file called SYSTEM.WRK.TEXT.

E, for E(xit, leaves the Editor without saving anything. Any changes or additions to the file are lost permanently.

R, for R(eturn, returns you to the Editor without saving your workspace in a file. Your text will still be displayed on the screen and you can edit it and save it later.

W, for W(rite, saves the file under whatever name you want to give it.

Let's create a file. Press **W** to write your workspace to a file.

The following prompt appears

```
Quit:  
Name of output file (<cr> to return)
```

Let's store your workspace in a file called "TOSAY" on volume #4. Type **#4:TOSAY** (if you make a mistake, press the left-arrow key to backspace and erase, then retype the file name).

When you have typed your file name correctly, press **RETURN**. As the Editor writes your file to disk, you see the word "Writing" followed by some periods. When your file has been stored, the following appears:

```
Your file is 42 bytes long  
Do you want to E(xit from or R(eturn to the Editor?)
```

You can Return to the workspace you just saved by pressing **R**, or Exit from the Editor by pressing **E**.

Press **[E]** to Exit from the Editor. You return to the Command promptline.

Your workspace is saved in a file on volume #4 called TOSAY.TEXT. The Editor automatically adds the extension .TEXT to any file you save or retrieve. Because it adds this automatically, you should never type .TEXT when saving or retrieving a file. (If, for example, you tell the Editor to retrieve a file called TOSAY.TEXT, it will add .TEXT to the name, search for a file called TOSAY.TEXT.TEXT, and fail to find it.)

If you want to take a break from this tutorial now, press **H** to Halt, and turn your computer off.

If you want to continue, press **[E]** to return to the Editor.

Lesson 2: More Advanced Instructions

You have pressed **E** to enter the Editor and it is asking what file you want to edit

```
Edit: No work file is present. File? ( <ret> for no file)
```

Let's continue working on the file **TOSAY**. To retrieve this file, type **#4:TOSAY** and press **RETURN**. You see the Edit menu and the text from your file on the screen.

I(nsert

When you insert words into existing text, first move the cursor to the character you want to insert in front of, and then select Insert.

Your text should look like this:

```
This is just
to say
I have
eaten
```

Move the cursor to the 'j' in 'just' and press **I**. The word 'just' jumps to the right margin and you see the following prompt:

```
>Insert: text {<bs> a char,<del> a line} [<etx> accepts, <esc> escapes]
```

Type `o n l y` and press the `SPACE BAR`.

The Insert promptline tells you that `<etx>` (that is, `^C`) accepts the insertion, while `<esc>` escapes the insertion. Press `^C`. This tells the Editor that you want to save your insertion. After you press `^C`, the word 'just' moves back from the right margin. If you had pressed `ESC`, the word you had typed would not be inserted into the text.

Your screen should look like this

```
This is only just
to say
I have
eaten
```

D(elete)

The first line looks awkward. It might look better if you deleted the word 'just'.

Move the cursor to the 'j' in 'just' and press `D`. You see the following prompt:

```
>Delete: < > <Moving commands> {<etx> to delete, <esc> to abort}
```

Press the `SPACE BAR` four times. Each time you press the `SPACEBAR`, a letter disappears from the screen. Press `^D` four times. Each time you press the left-arrow key, a deleted letter reappears.

You may wonder what has happened to your cursor-moving keys. The answer is that Delete uses these keys differently. The

location of your cursor when you enter Delete is called the "anchor." When you move away from the anchor, characters disappear. When you move toward it, deleted characters reappear. This allows you to change your mind while you're deleting.

Delete the word 'just' (if it's still on your screen). The prompt tells you to use "<etx> to delete" and "<esc> to abort". Let's save the deletion.

Press ^C. You return to the Edit menu. If you had decided you didn't want to delete the material, you would have pressed ESC instead of ^C.

F(ind and R(eplace

There are two commands in the Editor that search for material in a file: F(ind and R(eplace. These commands search for any letter, word, or phrase in your file. We will refer to the letter, word, or phrase you are finding or replacing as a phrase.

F(ind moves your cursor to the phrase you specify. R(eplace moves your cursor to the phrase, and then replaces it with another phrase.

Before you use F(ind or R(eplace, there are some conventions we will explain to you.

Repeating factors

The repeat factor you used with the cursor-moving keys can also be used with these commands. Specify the repeat factor before you press F or R to Find or Replace. This repeating factor has a different meaning for each of these commands. If you type 2 and then press R it means "find and replace two times." If you type 2 and then press F, it means "find the second occurrence of the phrase."

Move the cursor to the 'T' in 'This', type **[2]**, and then press **[R]**. The following appears on your screen.

```
>Replace[2]: L(it V(fy <targ> <sub> ->
```

The > at the start of the prompt means you are searching between the cursor position and the end of the file.

The number 2 inside the square brackets indicates that the Editor will replace the phrase twice.

Target and substitute strings

On the Replace promptline you see <targ> <sub>. When you use Find, you will see only <targ>. When searching for a phrase with the F(ind or R(eplace commands, the phrase you are looking for is called the target phrase. This is indicated by <targ>. When using the R(eplace command, the phrase you want to replace the target phrase is called the substitute phrase. This is indicated by <sub>.

Marking the phrase

You must mark the beginning and end of the phrase you are looking for (the <targ> phrase) and the phrase you are replacing it with (the <sub> phrase) with a marking character. For example, if you were using a slash as the marking character, you would type /fun/ to search for the word fun. You can use any character other than a letter or a number as a marker, but you must use the same character to mark both phrases.

In this manual, we will use the **[]** symbol as a marking character. (Don't confuse this with the infinite repeat factor. The marking character does nothing but separate words.)

Literal and token modes

When using the F(ind or R(eplace commands, you can specify whether you want to search for whole words or for parts of words.

The **literal** mode searches for any occurrences of the phrase, even if it is part of a word. The **token** mode searches for whole words only. A whole word is a word surrounded by spaces or other punctuation. For example, if you have the sentence "Put the book in the bookcase," and you are looking for "book", the literal mode would find two occurrences of "book", while the token mode would find only one—the word "book" surrounded by spaces.

In addition, the token mode ignores spaces within the target and substitute phrases. For example, the token modes considers, (" at ") and ("at") to be the same.

Verifying

When you are replacing phrases, you can ask to examine the phrase before it is replaced by using Verify. This is shown in the Replace promptline as V(fy.

We return to our example. You have typed 2 and then pressed R and you are looking at the Replace promptline.

```
>Replace[2]: L(it V(fy <targ> <sub> =>
```

We want to search for the letter 's', replace it with the letter 't', and verify before the replacement. Since the letter 's' is not a whole word, we want to search in the literal mode, so press . Press to verify the replacement before it occurs. We will use the character as our marking character. You would type the following.

[L]	for the literal mode
[V]	to verify
[/s[/[/t/]	to find the letter 's' and replace it with 't'

The promptline will look like this

```
>Replace[2]: L(it V(fy <targ> <sub> =>LV/s//t/
```

As soon as you press the last / after the letter 't', Replace begins its search.

When Replace finds the letter 's', it will ask you to verify by displaying the following:

```
>Replace[2]: <esc> aborts, 'R' replaces, ' ' doesn't
```

Press [R] to replace, press the [SPACE BAR] to leave the phrase as it is and go to the next phrase, or press [ESC] to stop replacing.

When you have found the letter 's' twice, you return to the Edit menu.

Searching For Previously-Used Phrases

You may reuse the last phrase you found or replaced by typing **S** instead of the phrase. (Don't type the marking character around the S.) For example, if you wanted to search for the letter 'a' and replace it with 't', you would type the following:

[R]	To replace
[L]	To search in literal mode
[/a/]	To find the letter 'a'
[S]	To replace it with the previous substitution (in our example, this is the letter 't'.)

After you press **S**, the Editor replaces the letter 'a' with the letter 't'.

Controlling the Margins

Move the cursor after the last character on your screen. Press **I** to Insert. Press **RETURN** twice. This starts a new paragraph. Now type the following and *don't* press RETURN until we tell you to. Don't worry if your screen behaves oddly.

I ate the grapes, the oranges, the apples, the pears, the bananas, the pineapples, and the cucumbers

When the text you are typing nears the right margin, you hear a beep. When it reaches the right margin, you see an exclamation mark. After that, you can't see the characters you are typing. The editor seems to be ignoring the right margin of your screen.

Now, press **RETURN** twice. Your cursor appears back on the screen. Type the above sentence again, but this time, press RETURN when you hear the beep. (If you are in the middle of a word, type **-** (hyphen) to split the word before you press RETURN.) When you finish typing the sentence, press **RETURN** twice. This time, you can see the entire sentence on the screen.

The editor has three ways of controlling the margins.

1. Start this line where I started the preceding line (this is called **auto-indent**) and don't start a new line until I press RETURN.
2. Start this line at the left margin (regardless of where I started the preceding line) and don't start a new line until I press RETURN.
3. Start this line at the left margin and start a new line when the text goes over the right margin (this is called **filling**).

The first way (auto-indent) is convenient for writing programs, outlines, or tables. The second way (no auto-indent or filling) is convenient for writing short lines lined up on the left margin. The third way (filling) is convenient for writing paragraphs.

The Editor is currently using the first way of controlling the margin. We want to use the third way (filling).

Press **[C]** to save your insertion and return to the Edit menu.

Press **[S]** (for Set) to change the way the Editor controls your margin.

Set allows you to change many of the things the Editor assumes. We will explain Set fully in the reference section. For now, we only want to change how the margins are controlled. The following prompt is on your screen:

```
Set: M(arker E(nvironment <esc>
```

Press **[E]** to set the Environment. You will see a long list of options. The only ones that concern us now are A(uto indent and F(illing):

```
A(uto indent   True
F(illing       False
```

To force the text between the right and left margin (the third way of controlling the margins) you want A(uto indent to be False and F(illing to be True. Press **[A]** and then press **[F]**. This turns auto-indent OFF (by setting it to False). Press **[F]** and then press **[T]**. This turns filling ON (by setting it to True). These two options should now look like this.

```
Auto indent   False
Filling       True
```

Press the **[SPACE BAR]** to return to the Edit menu. Move your cursor to the end of your file (this should be two lines past your last line of text). Press **[I]** to insert and type the following sentence. Don't press RETURN when the text reaches the right margin:

I also ate the radishes, the turnips, the lettuce, the carrots, the onions, and the freezer

When your text exceeds the right margin, it automatically moves down to the next line.

If you want to fix up the sentence that went over the right margin, press **^[C]** to exit from Insert. Move the cursor to any place within that line, and press **[M]** for Margin. This sentence now appears between the right and left margins. The Margin command reformats paragraphs for the current margins when you are using filling to control the margins.

This filling mode of controlling the margin will be used until you use Set to change it. If you save the workspace with filling ON, filling will still be on when you retrieve the file.

We are ready to leave the Editor. Press **[Q]** to Quit. Then press **[W]** to write the current workspace to a file on disk. The following prompt appears:

```
Quit:
'$' <ret> writes to #4 TOSAY.TEXT
Name of output file (<cr> to return)
```

You can save your changes to a new file by typing the file

name and pressing RETURN, or you can replace the original file (in this case, TOSAY) with your updated workspace by typing \$ and pressing RETURN. Let's replace our original file with our updated file.

Type `$` and press `RETURN`.

You see the following prompt:

Do you want to E(xit from or R(eturn to the editor?

Press `E` to exit from the Editor.

The Reference Section of this manual describes each Editor command in detail and will give you some more information on editor functions.

Reference Section

The Reference Section lists and explains the options of the Editor. There are also explanations of Editor functions and quick reference guides.

Accepting or Rejecting Commands

You have the chance to change your mind about many of the changes made in the Editor commands before they are made to your text. If you want the changes made to your text, press **^C** (on your screen, it tells you to press `<etx>`). If you decide you don't want the changes made to your text, press the **ESC** key. For example, if you delete a line by mistake while in the **D**(elete command, press **ESC** and your text will be restored.

Copy Buffer

The Copy Buffer temporarily holds a copy of the text which was most recently inserted or deleted.

Text is stored in the copy buffer by the following commands.

1. **D**(elete: When you delete text, it is saved in the copy buffer. Deleted text stays in the copy buffer regardless of which command (**^C** or **ESC**) you use to exit from Delete.

Because the deleted text stays in the copy buffer even when you decide not to delete it, you can use this method to copy text within a file. Press **D** and delete the text you want to copy, then press **ESC**. You can use **C**(opy **B**uffer to place that text anywhere you like. The original text remains unaffected.

2. I(nsert: When you press ^C, the inserted material is copied into the buffer. When you press ESC, the copy buffer is emptied.
3. Z(ap: If you use the Z(ap command, the zapped material is loaded into the buffer.
4. M(argin: This command causes the copy buffer to be left empty.

Entering and Exiting from the Editor

After you put the p-System disk in Drive A, press RETURN to load it and you will see the p-System Command line at the top of your screen. Press E for E(dit).

If you do not have a work file the Editor asks you what file you would like to edit. At this point enter a filename, or press RETURN to Edit the workspace as "noname". You may name the file when you Q(uit and W(rite it. If you do have a work file, that file is automatically read in by the Editor. You create a

work file when you use the Q(uit U(pdate option in the Editor or the G(et file option in the Filer. To clear the work file so that you can edit another file, go to Filer and S(ave your current work file. You may get the message: "Workfile is saved". Don't worry about this. Press N for N(ew to clear the work file.

NOTE: Work files are designed for program development and should not be used with this version of the p-System. The text you create in Edit is called "workspace" and resides in the computer's memory until you W(rite it to a filename.

When you are finished with the Editor, press Q to Q(uit. You will have four choices. Be aware that U(pdate SHOULD NOT BE USED since it creates a work file. Use only E(xit, R(eturn, and W(rite. When you Q(uit and then Exit, or Q(uit, W(rite, and Exit, you will be returned to the p-System command line.

Cursor Movement Keys

There are several cursor-moving keys you may use in E(dit:

	moves the cursor one character to the right
	moves the cursor one character to the left
	moves the cursor up one line
	moves the cursor down one line
SPACE BAR	moves the cursor one character to the right
TAB	moves the cursor to the next tabstop
RETURN	moves the cursor to the beginning of the line below

Cursor Movement Commands

P(age	moves the cursor one full screen of lines up or down
J(ump	moves the cursor to the beginning of a file, end of a file, or to a marker
F(ind	moves the cursor to the location of a sequence of characters that you specify

Direction Indicator

The direction indicator determines which direction several of the cursor movement keys or commands will move. The > in the first position on the prompt-lines is the direction indicator and can be changed to turn the cursor movement to the left by pressing <. This changes the cursor direction and affects SPACEBAR, TAB, and RETURN. You can P(age back in a file by reversing the direction indicator. The direction indicator can be set forward by typing ">", ".", or "+". It can be set backward by typing "<", ",", or "-".

Repeat Factors

A repeat factor causes a command to be repeated a specified number of times. Before you enter the first letter of an Edit command, you may enter a number to have the command repeated that many times, or a "/" to cause it to repeat as many times as is possible within the file. Repeat factors also affect the Edit cursor commands listed above. For instance, you may type a number before pressing an arrow key and the cursor will move that number of places automatically.

Editor Options

ADJUST - moves one or more lines of text horizontally.

COPY - copies text from a file on disk or from the copy buffer.

DELETE - deletes existing text.

FIND - finds a sequence of characters.

INSERT - allows creation of new text.

JUMP - moves the cursor to a marker or to the beginning or end of a file.

KOLUMN - adjusts columns closer together or further apart.

MARGIN - rearranges text in a paragraph to conform to margins defined in Set.

PAGE - moves the cursor the distance of one or more screen heights in file.

QUIT - leaves the Editor and allows you to save text to disk.

REPLACE - replaces a sequence of characters with another sequence.

SET -

ENVIRONMENT - allows entry of various global settings which affect the Editor.

MARKER - places an invisible marker in the file.

VERIFY - redisplay the screen to verify that it correctly reflects the file contents.

XCHANGE - allows insertion of new text by overwriting old text.

ZAP - instantly deletes a portion of text.

The Editor Commands Listed by Function

Editing:

Delete
Insert
Replace
Verify
Xchange
Zap

Formatting:

Adjust
Kolumn
Margin
Set - Environment

Moving within a file:

Find
Jump
Page
Set - Marker

Copying and moving text:

Copy

Leaving the Editor:

Quit

NOTE: The Set command
affects the following Edit
commands:

Adjust
Find
Insert
Jump
Kolumn
Margin
Replace

The Editor Commands

The rest of this section lists the commands alphabetically. The way each command appears on the screen is shown, and a description of the command and its functions is given.

A(djust

On the menu: **A(djust**

Press **A** from the E(dit menu. This displays the following menu:

```
>Adjust: L(just R(just C(enter <arrow keys> {<etx> to leave}
```

The A(djust command moves the line the cursor is on.

Pressing **L** justifies a line to the left margin, pressing **R** justifies it to the right margin, and pressing **C** centers the line between the margins.

Each time you press the right-arrow key, the whole line moves one space to the right. When you press the left-arrow key, the line moves one space to the left.

To adjust another line, press the up-arrow (or the down-arrow key); the line above (or below) the previously adjusted line will be automatically adjusted by the same amount.

All arrow key adjustments may be repeated as often as necessary.

Use the S(et E(nvironment command to alter the margins.

Press **^C** to exit from the A(djust command and save changes; pressing **ESC** won't work here.

C(opy

On the menu: **C(opy**

Place the cursor where you want the text copied to and press **C** from the E(dit menu. The following menu is displayed.

```
>C(opy: B(uffer F(rom file <esc>
```

Press ESC to leave without copying.

The C(opy command copies text into the file from one of two sources: the copy buffer (described at the start of this section), or another text file.

To copy from the copy buffer, press **B**. The text in the copy buffer appears at your cursor location.

NOTE: To move text into the copy buffer, use: I(nsert and **^C**; D(elete with **<esc>** or **^C**; or Z(ap.

To copy text from another file, press **F**. The following menu is displayed.

```
>C(opy: From what file[marker,marker]?
```

To copy an entire file, type the name of the file and press RETURN. The copied text will appear at the cursor location.

To copy part of a file, type: **filename[marker, marker]**. There must be markers set at the beginning and end of the text you wish to copy (these markers are set with the Editor's S(et command). You may use two markers, or the file's beginning or end as a marker. For example, if you had set a marker called **HERE** at the middle of the file, typing **filename[,HERE]**

would copy from the start of the file to the marker and typing **filename[HERE,]** would copy to the end of the file.

When C(opy is finished, the cursor moves to the end of the copied text.

D(elete

On the menu: **D(el**

To D(elete, place the cursor where you want to begin deleting text and press from the E(dit menu. The following prompt is displayed:

```
>Delete: < > <Moving commands> {<etx> to delete, <esc> to abort}
```

The D(elete command uses the cursor's position when you pressed **D** as an "anchor." As you move the cursor away from the anchor, characters disappear. Moving back toward the anchor restores the deleted characters. To accept the deletion, press **^C**; to escape without deleting, press **ESC**.

When you are in Delete, you can use all of the cursor-moving keys described earlier, including repeat factors and global direction.

When the deleted text is too large to store in the copy buffer, the editor displays the following warning.

```
There is no room to copy the deletion.  
Do you wish to delete anyway? (y/n)
```

Press **Y** to delete the material with no chance of recovering it, or press **N** if you don't want to delete the material.

The following is an example of using delete on the screen below:

```
This is just  
to say  
I have  
eaten
```

1. Move the cursor to the 'e' in 'eaten',
2. Press < (this changes the cursor direction to left).
3. Press **D** to delete.
4. Press **RETURN** twice. Each time you press **RETURN**, it moves the cursor up a line, and the text on the line disappears.
5. Now press **^C**. After deletion, the screen looks like this:

```
This is just  
eaten
```

The two deleted lines are stored in the copy buffer, and the cursor returns to its initial "anchor" position. If you wish, you may now use **C** (copy) to copy the two deleted lines to any other place in the file.

F(ind

On the menu: **F(ind**

To use the **F(ind** command, press **F** from the **E(dit** menu. The system will display the following prompt:

```
>Find[n]: L(it <target> =>  
or  
>Find[n]: T(ok <target> =>
```

The **F**(ind command finds the *n*th occurrence (indicated by the number in square brackets) of the phrase you enter. The search starts from the cursor position and moves backwards or forwards (depending on the global direction shown by the bracket at the beginning of the menu). The cursor stops at the position immediately after the phrase. To find the *n*th occurrence, enter the number before you press **F**. The default is number 1. Enter a `"/` to find the last occurrence.

As we described earlier (in Lesson 2 of the Editor Tutorial), you can search for whole words only (token mode) or for parts of words (literal mode). The default (the mode you will search in if you don't specify a mode) is set using the **S**(et command. The other mode appears as an option on the promptline. In the first promptline above, **T**oken is the default mode and **L**iteral can be chosen by pressing **L**.

When you enter the phrase you want to Find, it must be preceded and followed by marking characters. Marking characters are special characters such as `.`, `^` or `/`.

After you find a phrase and want to find the next occurrence, press **F** for Find and then **S** for Same. To find the last occurrence of a phrase, press `/`, **F** for Find, then either **S** for Same or enter the phrase.

If the Editor can't find the phrase, the following message is displayed.

```
ERROR: Pattern not in the file. Please press <spacebar> to continue.
```

The following paragraphs show how to use the F(ind) command for the screen below.

```
This is just
to say
I have
eaten
```

1. Start in the Edit mode.
2. Move the cursor to the beginning of the file and press F.
3. When the Find menu appears, type `/have/` (the `/` character is the delimiter around the phrase you are searching for).

The prompt with your response is shown below.

```
>Find[1]: L(it <target> =>/have/
```

The cursor jumps to the space after the 'e' in 'have' and you are automatically returned to Edit.

Insert

On the menu: **I**(nsert

Place the cursor where you want to insert text and press `I`. The Editor displays the following menu.

```
>Insert: Text {<bs> a char, <del> a line} [<etx> accepts, <esc> escapes]
```

Characters are inserted into the text at the cursor position.

Use these commands to make corrections:

left-arrow key deletes characters to left

^Q deletes a line of inserted text and moves cursor to the beginning of line

^U deletes a line of inserted text and moves cursor to end of preceding line

Insert will behave differently depending on whether you are using Auto-indent or Filling (these options are set using S(et E(nvironment)).

Using Auto-indent

If auto-indent is **ON(TRUE)**, pressing **RETURN** indents the next line the same amount as the line above it. If auto-indent is **OFF(FALSE)**, pressing **RETURN** starts the next line at the left margin on your screen.

Using Filling

If filling is **ON**, and auto-indent is **OFF** the text you type is forced between the right and left margins. When text exceeds the right margin, it is automatically moved to the next line. New lines are started on the left margin. Hyphenated words may be divided on the right margin.

You may change the indentation by pressing the **SPACEBAR** or the **left-arrow** key at the beginning of a line. The first line of a paragraph may be indented differently than the remaining text (see S(et Environment)).

Example 1: In the following example, auto-indent is ON. If you type this

```
ONE [RETURN]
[SPACE BAR] [SPACE BAR] TWO [RETURN]
THREE [RETURN]
[LEFT-ARROW] FOUR
```

You will see this on the screen

ONE	original indentation
TWO	indentation changed by <space><space>
THREE	<return> causes same auto-indentation
FOUR	<left-arrow> changes indentation from level of line above

Example 2: With filling ON and auto-indent OFF, typing "I really like arrow-keys" with a narrow margin would create the following on your screen:

I REALLY	Auto-returned when next word would exceed margin
LIKE ARROW-KEYS	Auto-returned at hyphen Level of left margin

Filling also adjusts the paragraph after an insertion. This doesn't affect any line beginning with the command character (see S(et), which is also considered to be the end of the paragraph.

You may S(et new margins and readjust a paragraph with the M(argin command if filling is ON and auto-indent is OFF.

If you insert text and press ^C, the inserted information is moved to the copy buffer. If you insert text and press ESC, the information is not moved to the copy buffer.

J(ump

On the menu: **J(ump**

When you press **J** for J(ump, the following menu appears:

```
>JUMP: B(eginning E(nd M(arker <esc>
```

Pressing **B** moves the cursor to the beginning of the file and **E** to the end of the file. Pressing **M** displays the following prompt:

```
Jump to what marker?
```

Markers are used to mark positions in your file. See the **M(arkers** command of the **S(et** command for more information. **J(ump** will jump to previously-set marker only.

K(olumn

On the menu: **K(ol**

Place the cursor on the line you want to move and press **K** for K(olumn. You will see the following menu.

```
>K(olumn: <vector keys> {<etx>, <esc> CURRENT line}
```

This command is used to move columns of text. **K(olumn** horizontally moves that part of the line that is to the right of the cursor.

The vector keys are the arrow-keys. The right arrow-key will move text to the right of the cursor one position to the right. The left arrow-key moves text to the right of the cursor one position left. Any text moving "into" the cursor will be lost.

The up and down arrow-keys move the lines above or below the same amount as the current line. If you move text horizontally in one of these lines, that movement will be "added on" to lines you vertically move to next. Press **^C** to leave **K(olumn**. If you press **ESC**, the last line you changed returns to its original alignment.

Note:

*When using **K(olumn**, pressing the left-arrow deletes the character above the the cursor. It's easy to do this and any characters deleted aren't saved in the copy buffer, so be careful when using **K(olumn**.*

M(argin

On the menu: **M(argin**

No promptline.

Move the cursor to the paragraph you want to adjust and press **M** for **M(argin**. For **M(argin** to work, filling must be **ON** and auto-indent must be **OFF** (see the **S(et** command). The paragraph is adjusted within the current margins. All the lines within the paragraph are justified to the left margin, except the first line, which is justified to the paragraph margin. You can set these margins with the **S(et E(nvironment** command.

The cursor may be anywhere within the paragraph when you press **M**.

The examples below show the margin settings and the paragraph adjusted within those margins.

Left-margin, 0
Right-margin, 40
Paragraph-margin, 8

This quarter, the equipment is different, the course materials are substantially different, and the course organization is different from previous quarters. You will be misled if you depend upon a friend who took the course previously to orient you to the course.

Left-margin, 8
Right-margin, 40
Paragraph-margin, 0

This quarter, the equipment is different, the course materials are substantially different, and the course organization is different from previous quarters. You will be misled if you depend upon a friend who took the course previously to orient you to the course.

A paragraph is any block of text surrounded by blank lines; lines beginning with a command character (see S(et)); or the beginning or end of the file. If the text file or the paragraph is especially long, the system may remain blank for several seconds while M(argin it working. When M(argin finishes, the paragraph redisplay. M(argin never splits a word, except at a hyphen.

Command characters

M(argin won't affect a line if the line starts with a command character. The command character must be the first nonblank

character in the line. M(argin treats lines beginning with the command character as blank lines. The command character itself is any character set using the S(et E(nvironment command.

P(age

On the menu: **P(age**

No promptline.

Pressing **P** from the Edit menu moves the cursor one screen forward or backward, depending on the global direction. If you press a number before pressing **P**, you move that number of screens.

Q(uit

On the menu: **Q(uit**

When you press **Q** for **Q(uit**, you see this menu.

```
>Quit:
U(pdate the work file and leave
E(xit without updating
R(eturn to the editor without updating
W(rite to a file name and return
```

Press **U**, **E**, **R**, or **W**.

U(pdate

Saves your text file in a working file called SYSTEM.WRK.TEXT and returns you to the Command promptline. Update SHOULD NOT BE USED. (It is for program development and you do not have the full development system.)

E(xit):

Returns you to the Command promptline without saving any of your changes, unless you have previously saved the file using the W(rite command (described below).

R(eturn):

Returns you to the Editor without saving your changes in a file. You may use this command if you pressed **Q** unintentionally. The text on your screen will be exactly as it was before you pressed **Q**.

W(rite):

If you have created a file from scratch (i.e. you are not editing a file that was already saved on disk), pressing **W** will display this menu:

```
>Quit:
Name of output file (<cr> to return) ->
```

You may now save your new file (workspace) and give it any valid filename (see the p-System Tutorial for information on filenames). Enter the **FILENAME** without **.TEXT** at the end. This can be preceded by a volume name or a disk drive number and a colon. Press **RETURN**. Your file will be saved as the filename you specified followed by **.TEXT**. You also have the option of pressing **RETURN** which will return you to your text in Edit without saving it to a file.

If the file you want to Write was originally read from an existing file, the W(rite command will display this menu:

```
>QUIT:
'$'<ret> writes to FILENAME.TEXT
Name of output file (<cr> to return) ->
```

You now have three choices:

1. Enter a new **FILENAME** and press **RETURN** to save your updated file to a new filename.
2. Enter **\$** and press **RETURN** to save your updated file to the same filename that it had originally and remove the old copy of the file. The original filename is displayed on your screen.
3. Press **RETURN** to return to your file in the Editor without writing to disk.

When you successfully **W**(rite a file, you will see the following on your screen:

```
>Quit:
Writing.....
Your file is 1978 bytes long.
Do you want to E(xit from or R(eturn to the editor?
```

Press **E** to exit from the Editor or **R** to return to the Editor and continue editing where you left off.

You may receive this Error Message when **W**(riting a file:

```
ERROR: Writing out the file Please press <spacebar> to continue.
```

This error message can occur for several reasons.

First: You have entered an incorrect filename. Make sure your filename is correct (do not add **.TEXT** - this will be done automatically).

Second: The Volume or Drive chosen is not on line. Check to see that the disk drive you are saving the file to is active and contains a p-System disk.

Third: There is not enough space on the disk to save your file. You may press the SPACEBAR and then W(rite the file to another disk. Check the p-System Tutorial for information on the Filer commands Transfer, Remove, and Krunch. These enable you to manipulate disk space and transfer files. This may allow you to transfer the file back to a disk you originally intended to save it on.

R(eplace

On the menu: **R(plc**

For more information on R(eplace, see the Lesson 2 in the Edit Tutorial.

When you press **R** for R(eplace, the following menu appears.

```
>Replace[1]:L(it V(fy <targ><sub> =>
      or
>Replace[1]:T(ok V(fy <targ><sub> =>
```

R(eplace finds the phrase you are looking for (indicated by <targ>) and replaces it with the phrase you want to substitute (indicated by <sub>).

To Replace more than one occurrence of a phrase, enter the number (repeating factor) before you press **P** for R(eplace. This number will appear in the brackets following the word Replace on the menu. Replace will find the number of occurrences you have specified and automatically exit you to Edit whether you replace the phrases or not. To find all occurrences of a phrase, enter / before pressing **R** for R(eplace.

As described earlier, you can search for whole words (Token mode), or for parts of words (Literal mode). You can search in the mode shown in the promptline (L(it or T(ok) by pressing **L** or **T**. If you don't specify the mode, you will search in the mode *not* shown on the promptline.

Pressing **V** for V(erify (shown on the screen as V(fy) allows you to examine each phrase found in the text so you can decide if it is to be replaced. When you ask to Verify, the following prompt is shown before each replacement:

```
>Replace: <esc> aborts, 'R' replaces, ' ' doesn't
```

Pressing **R** replaces the phrase and Replace searches for the next occurrence if you have entered a repeating factor. Pressing **SPACE** leaves the phrase as it was and Replace searches for the next occurrence of the phrase. You can press **ESC** to stop Replace.

If the target string can't be found, the following appears.

```
ERROR: Pattern not in the file. Please press <spacebar> to continue.
```

The cursor appears after the last phrase that was replaced.

Example 1:

Press **R** to start replace. When you see the Replace prompt, press **L** to search in the literal mode, and then type **/LOW//HIGH/**. Your promptline will look like this:

```
>Replace[1]: L(it V(fy <targ><sub>=>L/Low//High/
```

Low is the phrase you're looking for, **High** is the phrase you want to replace it with, and / is the marking character around the phrases.

This command will change:

"Lowly" to "Highly"

We used literal mode because the letters 'Low' are part of the word 'Lowly'.

Example 2:

To replace the whole word 'this' with the word 'that' twice. in the example below,

```
this is this and  
that is that
```

type **2** to replace twice, then press **R** to start R(eplace). You will see the following menu:

```
>Replace[2]: L(it V(fy <targ> <sub>
```

Type **V/this//that/** and the cursor will stop at each occurrence of "this". Press **R** each time and your screen will look like this:

```
that is that and  
that is that
```

Set

On the menu: **S**(et

When you press **S** for S(et, the following menu appears

```
>Set: M(arker E(nvironment <esc>
```

S(et E(nvironment

You can set the editing environment the way you want it. These settings will determine ways in which the Editor will behave. Press **E** from the Set menu to Set the E(nvironment. The following display appears:

```
>Environment: {options} <spacebar> to leave
A(uto indent   True
F(illing       False
L(eft margin   1
R(ight margin  80
P(ara margin   6
C(ommand ch    ^
S(et tabstops
T(oken def     True
```

```
3152 bytes used, 29612 available.
```

```
Editing: SCHEDULE.TEXT
Created March 10, 1982; last updated March 24, 1982 (revision 0)
Editor Version [IV.1 F6c].
```

This screen displays several options which are described below. Below the list of options are some important pieces of information.

The number of bytes used in your file and the number of bytes still available are displayed. The name of the file you are editing is displayed. The dates the file was created and last

updated are shown. The revision number indicates how many times the file has been updated to disk. The Editor version number is also displayed.

If you have used **F**ind and **R**eplace, the target and substitution strings will be shown on the screen as "Patterns:". Also, any markers you have used will be shown after "Markers:".

By pressing the appropriate letter, you may change the options. Any changes you make in **S**et will only affect the current file and will be saved until you reset them.

Environment Options

***A*(uto indent:**

Auto-indent causes the next line to have the same indentation as the current line when you are in **I**nsert. Refer to the section on **I**nsert. Auto-indent is turned **ON** by typing **AT** (for Auto-indent True) and turned **OFF** by typing **AF** (for Auto-indent False). Auto-indent True is the default.

***F*(illing:**

Filling allows paragraphs to be rearranged by **I**nsert and **M**argin. The lines in the paragraphs are made as wide as is possible within the margins (defined below). Filling must be True and Auto indent False for this to occur. (Refer to the **S**et and **M**argin sections.) Filling is turned **ON** by typing **FT** (for Filling True) and turned **OFF** by typing **FF** (for Filling False). Filling False is the default.

***L*(eft margin, *R*(ight margin, *P*(ara margin:**

The margins you set here will work with **I**nsert and **M**argin when you have also turned **F**illing **ON** and **A**uto indent

column number changes as you press the right or left arrow keys or by pressing **C**, entering a column number, and then pressing RETURN.

Press **T** to insert or delete a tab. When you press **T** the **T** in the line above will change to a dash. Pressing **T** again changes the dash back to a **T**.

T(token def:

This option affects **F**(ind and **R**(eplace. To search for whole words only, turn Token **ON** by typing **TT** (for Token True). Token def is True by default. To search for parts of words, turn Token **OFF** by typing **TF** (for Token False). When Token is **ON**, **F**(ind and **R**(eplace search for tokens by default. When it is **OFF**, **F**(ind and **R**(eplace search for literal strings by default. (see the discussion in "Lesson 2" of the Editor Tutorial.)

S(et M(arker

When editing, it is convenient to jump directly to certain places in a long file by using markers. Once a marker is set, you can jump to it by using the **M**(arker command in **J**(ump. Markers can also be used by **C**(opy **F**(ile to copy a portion of a File.

Move the cursor to the place in your text where you want the marker, press **S** for **S**(et, and press **M** for **M**(arker. The following prompt appears:

Set what marker?

Give the marker a name (up to eight characters) and press RETURN. The marker will be set at the cursor position and will be invisible. If you use the name of a marker that already exists, the original marker will be removed and replaced with the new one.

You can set up to 20 markers. If you try to set more than 20 markers, the following message appears.

```

Marker ovflw. Which one to replace?
(Type in the letter or <sp>)
a) name1      b) name2      c) name3      d) name4
e) name5      f) name6      g) name7      h) name8
i) name9      j) name10     k) name11     l) name12
m) name13     n) name14     o) name15     p) name16
q) name17     r) name18     s) name19     t) name20
  
```

Press a letter ("a" through "t") and the marker you are setting will replace that existing marker, or press **SPACE BAR** to return to Edit without replacing a marker.

V(erify)

On the menu: **V(erify)**

Causes the screen to redisplay to show the actual contents and remove any extraneous characters.

X(change)

On the menu: **X(change)**

Move the cursor to the text you want to exchange and press **X** for X(change). The following menu appears:

```
>eXchange: Text <vector keys> {<etx>, <esc> CURRENT line}
```

When you type, Xchange replaces the characters on the screen with the characters you are typing. While in X(change), typing **^X** inserts one space at the cursor's location, and **^Z** deletes a simple character at the cursor location.

For example, with the cursor under the 'u' in 'just' press X to exchange.

```
This is just  
to say
```

If you type **et**, and press the `SPACE BAR`, the screen will show:

```
This is jet  
to say
```

Pressing **^C** saves the changed line and returns you to the Edit menu. Pressing **ESC** restores the original text only in the last line you changed.

You may only exchange text to the right of the cursor while you are typing. To move the cursor, use the **arrow-keys**, **RETURN**, and **TAB**.

Z(ap

On the menu: **Z(ap**

Z(ap deletes from the first character of the last **F**(ind, **R**(eplace, or **I**(nsert operation to the current cursor position. If more than 80 characters are being zapped, the editor asks for verification.

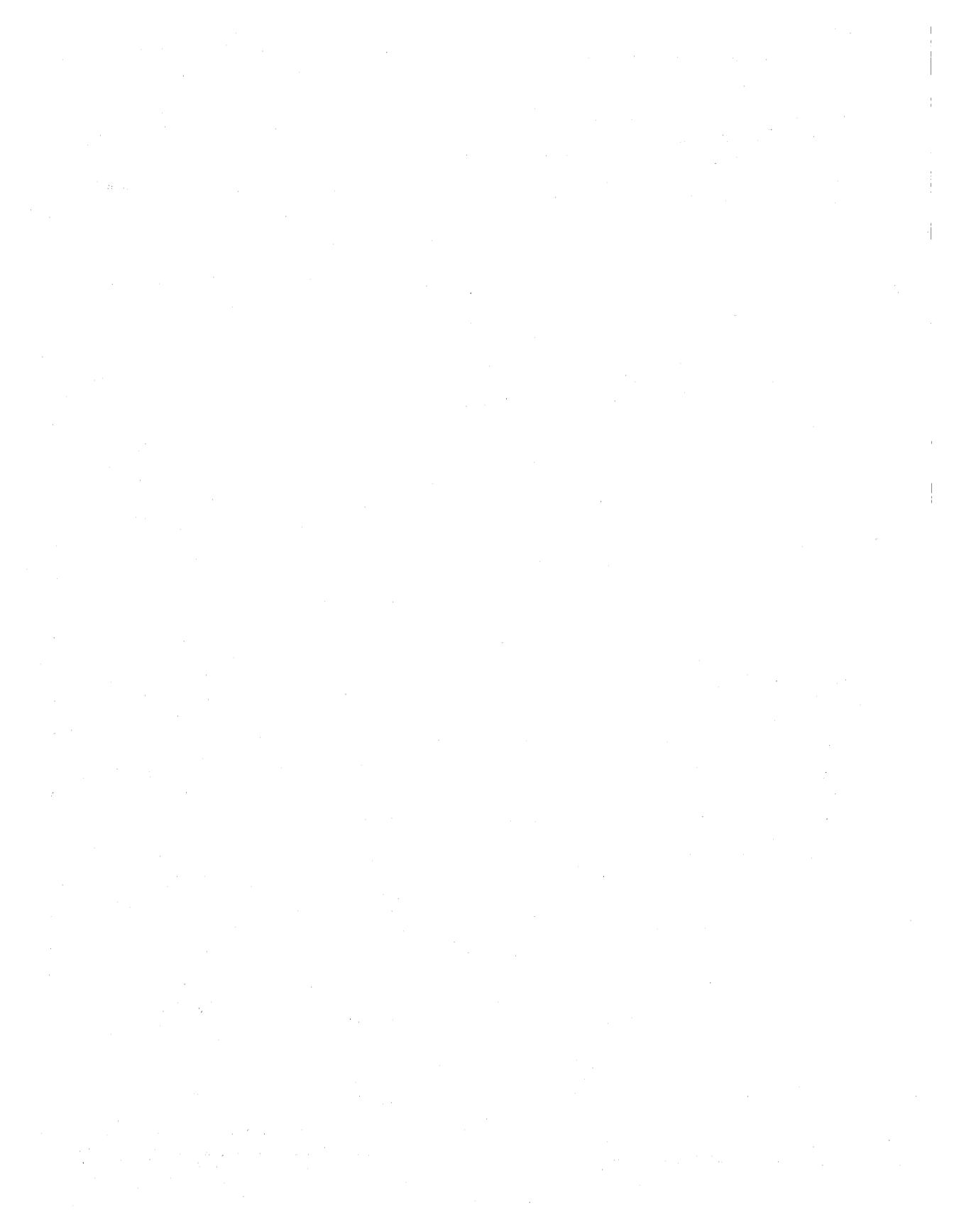
Before a **Z(ap**, you can determine the first character of the previous **F**(ind, **R**(eplace, or **I**(nsert by pressing the equal sign.

To remove a large section of text, place the cursor at the beginning and **I**(nsert a space. Then move the cursor to the end of the text to be deleted and press **Z** for **Z(ap**.

Whatever you deleted with the Z(ap command is stored in the copy buffer and can be retrieved with C(opy. If there isn't enough room in the copy buffer, you are asked if you want to Z(ap anyway. If you press Y for yes, you will not be able to recover the text deleted by Z(ap.

Z(ap isn't allowed after using A(djust, D(elete, K(olumn, M(argin, or a previous Z(ap.

SECTION 3
p-System
Print



Introduction

PRINT prints the text files that you create with the p-System Editor. With PRINT you can break a document into pages and put headings, including page numbers, on each page. You can also control the line spacing and the top and bottom margins of the printed text.

PRINT works with a wide variety of printers and can be used with either continuous form paper or with single-sheets of paper.

The following section describes the simplest use of PRINT. You may never need to know more. If you do, read the rest of this section for a description of all of PRINT's features.

Before you use PRINT, make sure that your system disk is properly configured for your printer (using UTIL's Configuration option).

Simple Uses of PRINT

To use PRINT, press , to: eX(ecute, from the Command promptline. Then type and press RETURN. PRINT shows a list of options.

PRINT automatically displays the most common printer settings. The most important of these settings are marked with a number in the list below and explained in order below.

Print [A2]: Select an option (type '?' for help):

```

I(nput->
O(utput-> PRINTER:
G(o. Print the input file on the output.
A(dvance. Skip to next page on the output.
M(ake script file for setting current parameters.
Q(uit. Leave this program.

```

No D(ouble-space the lines?

No N(umber the lines?

¹ No S(top before each page for single sheet loading?

² Yes U(se ASCII formfeed characters between pages?

1 F(irst page number

1 T(op margin size in lines

3 B(ottom margin size in lines

³ 66 P(age size in lines (total: includes margins and heading)

\ E(scape sequence flag character

. C(ommand line flag character

H(eader-> Page \page. File is '\file'. Printed on \date.

¹ There is continuous feed paper in your printer (rather than single sheets).

² Your printer advances to a new page when sent an ASCII "form feed" control character.

³ Each page has at least 66 lines.

If these built-in choices meet your needs, using PRINT is simple. If you are not sure of the settings for your printer, follow the instructions below to print a sample text file.

From the Print menu:

1. Press for I(nput. When asked for the filename, type the name of your text file and press . You will now see your filename in the PRINT menu.
2. Press for G(o to start printing.

PRINT will now print your text file.

3. If you need to advance to the end of the page to tear off the printout, press **A** for A(dvance).
4. When you are finished printing, press **Q**, for Q(uit, to leave PRINT and return to the Command promptline. Typing Q when you are printing immediately stops the printing.

For example, if we wanted to print the file TEST.TEXT on volume #4, we would do the following:

1. Press **X**, for eX(ecute a program, from the Command promptline.
2. Type **P R I N T** and press **RETURN** when asked what program you want to execute.
3. When the PRINT menu appears, type **I** to name the input file. When asked for the filename, type **T E S T . T E X T** and press **RETURN**.
4. Press **G** for G(o. This will print your file.

If the printout is the way you want it, this is all you need to know about PRINT.

If you want to change some of the PRINT options, read the rest of this section.

PRINT Options

To select or change a PRINT option, press the letter indicated in the menu. For example, to select Q(uit you would press **Q**.

You can also press **?** for more information about PRINT.

Many of the PRINT options display a prompt on the first line of the screen. Error messages are also shown on this line.

Controlling the Page Layout

Top margin, bottom margin, and page size

1	T(op margin size in lines
3	B(ottom margin size in lines
66	P(age size in lines (total: includes margins and heading)

There are three options which control the general page layout: P(age size, T(op margin, and B(ottom margin.

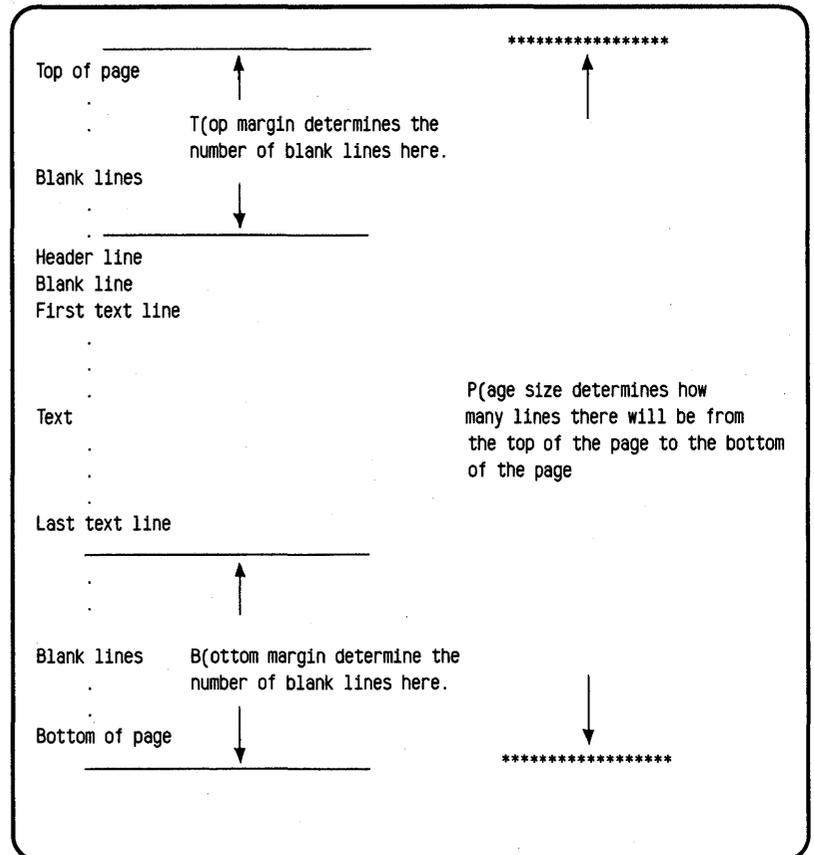
P(age size is the number of lines from the top of the page to the bottom of the page, including the top and bottom margin. To change page length, press **P** and type the number of lines.

T(op margin sets the number of blank lines between the top of the page and the heading. The heading is always *followed* by a blank line. To change the top margin, press **T** and type the number of lines you want. To eliminate the top margin, just press RETURN.

B(ottom margin sets the number of blank lines between the last line of text and the bottom of the page. To change the bottom margin, press **[B]** and type the number of lines you want. To eliminate the bottom margin, just press RETURN.

PRINT doesn't control the horizontal layout of text. The text is printed exactly as it appears in the file.

This is a diagram of a page with these settings indicated:



H(ader and F(irst page number

```

1 H(ader-> Page \page. File is ''\file''. Printed on \date.
  F(first page number

```

H(ader

H(ader sets the header line that is printed at the top of the page. The standard header line contains a page number, the name of the file, and the current date (the date is set using the Command promptline's D(ate command). Here is an example header line in the standard format:

```

Page 3. File is ''TEST.TEXT''. Printed on June 4, 1983.

```

To change the format of the header line, press **[H]**, type the new header line, and press **[RETURN]**. If you want to include the page number, filename, or date on the header line, read the instructions below. If you don't want a header line, just press **[H]** and press **RETURN**.

Escape sequences To tell the **PRINT** option to retrieve information and print it in the header line, you will want to include the following escape sequences in your header line: page, date, and file. We will first explain what information these commands retrieve, and then we will describe the format of these commands.

Page When this is included in your header line, the page number will be printed on each page.

File When this is included in your header line, the file name will be printed on each page. This will print the file name of either the main file, or the included file, whichever is being printed. (We will explain included files later.)

Date When this is included in your header line, the date will be printed on each page. (This date is set with the D(ate option on the Command promptline.)

Format for escape sequence To print the page, file, or date in the header line, type the current escape flag character and then type the information you want retrieved. In our example, the escape flag character is a back slash. This appears on the main PRINT menu as

\ E(escape sequence flag character

An example

Press **[H]** to create a new format for the header line. The text on the header option line disappears and you can type in a new format. If you want the heading on each page to include the title "p-System" and the date, you would type the following:

p **-** **S** **y** **s** **t** **e** **m** **.** **[SPACE BAR]** **** **d** **a** **t** **e**

and it would create this heading on your printout

p-System. February 16, 1983

If you wanted to add the page number to your current header line, you would type

p **-** **S** **y** **s** **t** **e** **m** **.** **[SPACE BAR]** **** **d** **a** **t** **e** **.** **[SPACE BAR]**
P **a** **g** **e** **[SPACE BAR]** **** **p** **a** **g** **e** **.**

on the H(ader option line and it would create this heading on your printout

p-System. February 16, 1983. Page 1

Changing the escape flag If you want to change the escape flag (in our example, a back slash) to another character, press **E** from the PRINT menu. You can now enter a new escape flag. For example, if you wanted the new character to be an asterisk (*), you would type ***** after you press **E**. The escape flag option line will look like this

```
*      E(escape sequence flag character
```

If you change the escape flag, preface the escape sequences with this new character. In our example, we would now type ***date**, ***file**, or ***page** to include this information in our header line.

In the next section we will explain how to include these escape sequences in your text file.

***F*(first page number**

The first page of the printout will usually be numbered Page 1. If you want the first page of your printout to have a different number, use the **F**irst page number option of the PRINT menu. For example, if you wanted the first page to be numbered Page 5, you would press **F** from the PRINT menu, and then type 5 and press RETURN. The following pages would be numbered consecutively. See the discussion above for including this page number in the header line.

***D*(ouble spacing and *N*(umbering lines**

```
No      D(ouble-space the lines?
No      N(umber the lines?
```

D(ouble spacing puts a blank line between each line of text on your printout. To select double spacing, press **D** from the

PRINT menu. Then type **Y** for yes and press **RETURN**. This will change "No" to "Yes" at the start of the option line. To turn off double-spacing, press **D** again. Then type **N** for no and press **RETURN**.

N(umber the lines puts a line number next to each line on your printout. It will not number the blank lines produced by double-spacing your printout. To turn ON line-numbering, press **N** from the PRINT menu. Then type **Y** for yes and press **RETURN**. This will change "No" to "Yes" at the start of the option line. To turn off line-numbering, press **N** again. Then type **N** for no and press **RETURN**.

Other PRINT Options

```
I(nput->
O(utput-> PRINTER:
G(o. Print the input file on the output.
A(dvance. Skip to next page on the output.
M(ake script file for setting current parameters.
Q(uit. Leave this program.
No      S(top before each page for single sheet loading?
Yes     U(se ASCII formfeed characters between pages?
.
C(ommand line flag character
```

We will discuss the rest of the PRINT options later.

Printer Commands Within the Text File

As mentioned above, PRINT prints text files as they appear on the screen.

However, you can include commands *within* your text file that will alter the printout. These commands within your text file can be escape sequences (described earlier), or command lines.

Escape Sequences Within Your File

You may include escape sequences to print the date, the name of the file, or the page number within your text file. To include this information in your file, type the escape flag character (for our example, a backslash) and then type **date**, **file**, or **page** to include the current date, file name, or page number. These escape sequences can be used within the body of the text, or within command lines. The escape flag can also be changed within the file by using a command line.

Command Lines

A command line is a line that alters the printout from within the text file. To include a command line in your text file, you must first type the command line flag character. If you look at the example PRINT option menu, you will see that the command character is a period (.). This means that every time PRINT sees a period as the first character on a line, it looks at the next two characters for the command to perform. If it finds

the command after the command character, it will follow it. If it doesn't find a command after the command character, the line is ignored. (You can place comments in your text by putting the command character at the start of the comment line.) When writing comments, leave a space between the command character and the comment.

Setting the command character

If your text lines begin with a command character, they will not be printed. You may want to change the command character if the you have text lines that begin with that character. For example, if you have lines of text that begin with a period, you might want to change the command character to an @ sign.

There are two places to change the command character: in the PRINT menu and within the text file. This should be the same command character you S(et in the Editor. Read the section on the Editor to set the command character there. To set the command character from the PRINT menu, press C and then type the new character. This new character will now be displayed in your menu. For example, to change the command character to an @ sign, press [C] from the PRINT menu, and then press [@]. Your new command character appears on the menu. To change the command character from within a text file, read the instructions below.

Commands

Command lines within a text file control the following: including another file, starting on a new page, changing the heading, changing the command character, changing the escape flag, and ending the file. These are explained in detail below. Command lines consist of: the command character, a command instruction, and, in some cases, information to be included in the command. You can abbreviate the command

instruction to two letters, upper- or lowercase. PRINT only reads the first two characters after the command character.

Information to be included in the command should be separated from the command instruction by at least one blank space and enclosed in single or double quotation marks.

The commands are:

INCLUDE To include another file in your printout, use the INCLUDE command. When PRINT finds this command in a file, it will print the file specified by the command, and then return to the original file and continue its printing where it stopped. All pages will be numbered consecutively. The file specified in the INCLUDE command cannot contain any INCLUDE commands.

The INCLUDE command line contains: the command character, the command INCLUDE (which may be abbreviated IN), a blank space, and the name of the file to include (enclosed in quotation marks).

For example, if you were printing a file that looked like this:

```
This is just  
.INCLUDE '#4:TOSAY.TEXT'  
eaten
```

and the file you were including (in this example, a file called TOSAY.TEXT on volume #4) looked like this:

```
to say  
I have
```

your printout would look like this:

```
This is just  
to say  
I have  
eaten
```

INCLUDE also allows you to store a large document in several files and print them as one file. To do this, add a list of INCLUDE commands to the main file. For example, if the first part of your document were in MYFILE1, the second part in MYFILE2, and the third part in MYFILE3, you would add the following commands to the end of MYFILE1.

```
.INCLUDE "#4:MYFILE2.TEXT"  
.INCLUDE "#4:MYFILE3.TEXT"
```

This will print the three files as one long file.

PAGE The PAGE command starts a new page in your printout. This is useful when the page breaks inserted by PRINT aren't the page breaks that you want.

The PAGE command contains: the command character and the command PAGE (this may be abbreviated PA). For example, to start a new page with the line "I have", you would type the following:

```
This is just  
to say  
.PAGE  
I have  
eaten
```

HEADING You can change the header line from within a text file with the HEADER command. The header line can also be changed from the PRINT menu.

The **HEADER** command allows you to print a specific heading for that document, or to have different headings on each page. You could, for instance, specify a blank heading on the first page or change headings with each chapter or file change.

The **HEADER** command contains: the command character, the command **HEADER** (this may be abbreviated **HE**), a blank space, and the header line (enclosed in quotation marks). You can also include an escape sequence in this heading.

For example, if we wanted the heading for the first chapter of the text file to include the chapter title "p-System Introduction" and the page number, we would type the following at the start of the first chapter:

```
.heading "p-System Introduction \page"
```

Notice that we have included the escape sequence `\page` in this header to print the current page number on each page.

If we wanted the header for the second chapter to include the chapter title "p-System Detailed Instructions", the page number and the date, we would type the following at the start of the second chapter:

```
.heading "p-System Detailed Instructions \page \date"
```

COMMAND You can change the command character (the character used to start a command line) with this command. You will want to do this if the current command character is used in your text. For example, if your current command character is a period and you want to print the following decimal numbers in your file:

```
.32  
+  
.65  
.97
```

you would want to change the command character to something other than a period.

This command contains: the current command character, the instruction COMMAND (which can be abbreviated CO), a blank space, and the new command character (enclosed in quotation marks.)

To change the command character from a period to an asterisk, you would type the following:

```
□ □ C O M M A N D □ SPACE BAR □ " * "
```

END The END command stops printing from the current file. If you put an END command in the main file, you will return to the PRINT menu when it reaches this command. If you put an END command in an included file, it will stop printing the included file and continue printing the main file where it left off.

This command is convenient if you want to keep some information in a file but do not want this information printed.

This command contains: the command character and the instruction END (this can be abbreviated EN).

For example, if you wanted to include some information about the status of the file within the file itself, you would type the following at the end of your file

```
.END  
This file still needs to be checked for spelling
```

This instruction would remain in the file, but would not be printed.

ESCAPE You can change the escape flag from within a file using the ESCAPE command. For example, if your current escape flag were a back slash, \, and you had the following lines in your text:

```
either red\blue or green\yellow
```

you would want to change the escape flag to something else.

The command contains: the command character, the instruction ESCAPE (which can be abbreviated ES), a blank space, and the new escape flag (enclosed in quotation marks). For example, if you wanted to change the escape flag to &, you would type the following:

```
[ ] [E] [S] [C] [A] [P] [E] [ ] [SPACE BAR] ["] [ & ] ["]
```

Printing Your File

We will now discuss the remaining PRINT options:

```
I(nput-->
O(utput--> PRINTER:
G(o. Print the input file on the output.
A(dvance. Skip to next page on the output.
M(ake script file for setting current parameters.
Q(uit. Leave this program.

S(top before each page for single sheet loading.
U(se ASCII formfeed characters between pages.
```

I(nput file

```
I(nput-->
```

To print a text file, you must first tell PRINT what file you want to print. You do this with the I(nput option. Press from the print menu. When asked for the filename, type the volume number and the file name, and press . For example, if you wanted to print a file called MYFILE on volume #4:, you would press from the print menu, and then type and press . The I(nput line would now look like this:

```
I(nput--> #4:MYFILE
```


On many printers that allow single sheets of paper, the paper must be inserted about an inch past the printing mechanism. If you're using such a printer, you may want to reduce the P(age size and possibly change the T(op margin, as well. For instance, if your printer prints 6 lines per inch and you're using standard 11-inch paper, you might reduce the P(age size from 66 lines to 60 lines.

U(se **ASCII formfeed** automatically advances the paper to the next page. Most printers can use this ASCII form feed character. If your printer can't, turn off the **U**(se form feed option. The effect will be the same as a form feed (only slower), as long as **PRINT**'s page size and margin options are properly set. To turn **U**(se off, press **U**. Then press **N** for no and press **RETURN**. The "Yes" at the start of the line changes to a "No." To turn it back on, press **U** again. Then press **Y** for yes and press **RETURN**.

A(dvance the printer

A(dvance. Skip to next page on the output.

You can advance the paper to the next page by pressing **A** from the print menu. You may want to do this before printing if your paper is halfway down the page and you want to start printing the file on the next page.

G(o

G(o. Print the input file on the output.

To start printing your file, press **G** for **G**(o. If it doesn't print your file, make sure that your printer is properly hooked up to

your computer and that it is turned on. You must also have configured the boot tracks on your system disk for your printer (read the section on the UTIL program to do this).

Q(uit)

Q(uit. Leave this program.

To exit from the PRINT menu and return to the Command promptline, press **Q**.

M(ake a script file)

M(ake script file for setting current parameters.

Once you become familiar with PRINT, you may find that there are certain options you usually have to change. To set these options automatically, M(ake a script file.

M(ake a script takes the the options you have set and stores them in a file. You may also include the command to eXecute PRINT in this file. If you do this, you can make the PRINT menu appear with your options already set.

To use M(ake a script, set the options in the PRINT menu the way you want them, and then press **M**. You will see the following prompt:

Enter name of script file:

Type the name of the script file you want to create and press **RETURN**. You can make this a text file by adding .TEXT to the end of the file name. The advantage of a .TEXT file is that it can be easily examined or modified by the p-System Editor. A

disadvantage is that it is at least four block long, whereas a typical script file is only one block long. When you type the script file name, you will see the following:

Enter name for invoking print:

Type the command you used to eXecute PRINT and press RETURN. In our case, we eXecute PRINT by typing **PRINT** from the eXecute prompt.

PRINT now creates a script file with your settings.

For example, if you were going to be revising a document text file called "SYSTEM" for a few weeks, you might want to have a script file for it. To create the script file, you would start PRINT and set the options as follows:

Print [A2]: Select an option (type '?' for help):

```

I(input--> SYSTEM
O(utput--> PRINTER:
G(o. Print the input file on the output.
A(dvance. Skip to next page on the output.
M(ake script file for setting current parameters.
Q(uit. Leave this program.

Yes D(ouble-space the lines?
No N(umber the lines?
Yes S(top before each page for single sheet loading?
No U(se ASCII formfeed characters between pages?

1 F(irst page number
1 T(op margin size in lines
3 B(ottom margin size in lines
66 P(age size in lines (total: includes margins and heading)

\ E(scape sequence flag character
. C(ommand line flag character

H(eader--> Page \page. File is '\file'. Printed on \date.
```

Once you have the options set, press **M** for M(ake a script. We want to create a script file called "SYSTEMSCRIPT" and store it on volume #5. When the first prompt

Enter name of script file:

appears, we would type

#5:SYSTEMSCRIPT and press **RETURN**.

In answer to the second prompt

Enter name for invoking print:

we would type **PRINT** and press **RETURN**. PRINT now creates a script file called SYSTEMSCRIPT and stores it on volume #5.

Once we had created this script file, we would eXecute it by pressing **X** from the command promptline. When asked for the name of the file to eXecute, we would type

=#5:SYSTEMSCRIPT and press

RETURN. The PRINT menu appears with the settings we used in our script file. To print the file the way you want it, just press **G** for G(o).

If you change the name of the PRINT program, or if you are not sure what volume the program is going to be on, you may not want to include the name of the PRINT program in your script file. When M(ake a script asks you for the name used to invoke print, press **RETURN** instead of typing the program's name. If you do this, you can eXecute your script file and your program by pressing **X** from the command promptline. When asked for the name of the file to execute, type the name of your printing program, **SPACE BAR I=#**, and the name of your script file. For example, if your print program (named

PRINT) were on volume #5 and your script file (named TEST) were on volume #4, you would type the following:

```
# 5 : P R I N T [SPACE BAR] I = # 4 : T E S T
```

and press `RETURN`.

This will automatically load your printing program with the settings in your script file. Press `G` to start printing.

Summary of Menu Items

By selecting any of the options below, you can:

I(nput	Choose the file to be printed.
O(utput	Change the destination of the print operation.
G(o	Print the input file to the output, according to the current option settings.
A(dvance	Skip to the next page on the output.
M(ake script	Build a script file which will invoke PRINT with the current option settings.
Q(uit	Leave PRINT.
D(ouble space	Select single- or double-spaced output.
N(umber	Cause each line to be preceded by a line number for the current page.
S(top	Specify whether single sheet loading or continuous forms are assumed by PRINT.
U(se ASCII FF	Specify whether the form feed character or a sequence of empty lines is used to separate output pages.

F(irst page	Specify the starting number on the first page of a document.
T(op margin	Specify the number of blank lines between the top of the page and the header line.
B(ottom margin	Specify the number of blank lines between the last line of text and the bottom of the page.
P(age size	Specify the total number of lines per page including the top and bottom margin.
E(scape	Change the character which starts an escape sequence.
C(ommand	Change the character which starts a command line.
H(earer	Specify the contents of the heading line at the top of each printed page.

Summary of Command Lines

By using the following commands, you can:

INCLUDE	Insert an additional file into the document being printed in place of the include command.
PAGE	Cause an immediate page break.
HEADING	Specify the contents of the heading for subsequent pages.
COMMAND	Change the command line flag character.

ESCAPE	Change the escape sequence flag character.
END	Terminate printing the current text file.

Summary of Escape Sequences

When any of the following is preceded by the escape sequence flag, the indicated text is substituted:

PAGE	The current page number.
FILE	The current input file name.
DATE	The current calendar date as maintained by the p-System.

Index

CPM Plus

assembling a program, 65
 baud rates, 51
 built-in utilities, 5
 device assignments, 45
 DEVICE command, 45
 DIR command, 28
 diskette attributes, 21
 diskette label, 21
 diskette passwords, 24
 displaying device assignments, 50
 ED command, 52
 file passwords, 25
 GET command, 56
 HELP command, 9
 LINK command, 70
 logical devices, 43
 MAC assembler, 64
 merging files, 37
 passwords, 23
 physical devices, 43
 PIP command, 34
 PIP external device options, 41
 protocols, 43
 PUT command, 56
 PUT command options, 60
 RMAC assembler, 64
 SET command, 21
 SETDEF command, 18
 SHOW command, 28
 SID (Symbolic Instruction Debugger), 72
 SUBMIT command, 56
 SUBMIT files, 56
 system clock, 11
 System files, 27
 time-stamping, 14
 transient utilities, 5
 user numbers, 30
 wildcard characters, 37
 XON/XOFF protocol, 51

p-System Tutorial

\$, 107
 *, 102
 2WORD.INTERP, 145
 4WORD.INTERP, 145
 =, 107
 ?
 undisplayed options, 88
 wild card, 107

 application programs
 copying, 142
 floating point arithmetic, 146
 purchasing, 141
 running, 142

 B(ack-up, 139
 B(ad Blocks, 121
 BAD files, 99
 baud rate
 setting, 138
 blocks
 bad, 121, *see also* B(ad Blocks
 determining number, 95
 recovering bad, 122
 size listing, 110
 Universal Medium, 144
 boot tracks
 configuring, 137
 creating, 136

 C(hange, 111
 CODE files, 99
 recovering, 126
 C(onfigure, 137
 CONSOLE:, 100, *see also* screen
 displaying file, *see* T(ransfer
 COPYDUPDIR, 129, 130

Index

p-System Tutorial

- copying diskettes
 - B(ack-up, 139
 - T(ransfer, 115
- copying files, 113, *see also* T(ransfer
- data files, 99
 - recovering, 126
- D(ate, 90
 - files listing, 110
 - setting, 90
- default volume, 103
 - listing, 112
 - selecting, 117
- Development System, 81
- directory
 - creating, 94
 - duplicate, *see* duplicate directory
 - extended, 110
 - listing, 108
 - recovering, *see* COPYDUPDIR
- disk name, *see* volume names
- diskette
 - copying, *see* copying diskettes
 - format, *see* format
- DUMMY files, 126
- duplicate directory
 - MARKDUPDIR, 128
 - Z(ero, 94
- E(xamine, 122
 - K(runch and, 125
- eX(ecute
 - application programs, 142, 144
 - COPYDUPDIR, 129
 - MARKDUPDIR, 128
 - M(onitor file, 133
 - UTIL, 92, 135
- EXECUTION ERROR #11, 145
- EXECUTION ERROR #17, 145
- E(xtended Directory, 110
- file names, 97
 - changing, 111
- file types, 98
 - BAD, 99
 - CODE, 99
 - data, 99
 - defaults, 100
 - TEXT, 99
 - WRK, 99, 118, *see also* work files
- Filer, *see also* (individual Filer options)
 - leaving, 105
 - options, 104
- files
 - copying, *see* copying files
 - crunching, 124, *see also* K(runch
 - printing, *see* printing files
 - recovering, *see* recovering files
 - removing, 111
- F(lip-Swap/Lock, 105
- floating point arithmetic, 144
- F(ormat, 92
- format
 - Osborne, 93, 141
 - Universal Medium, 141
- formatting a diskette, *see* F(ormat
- FOUR.BOOT, 146
- G(et, 118
- H(alt, 132
- I(nitialize, 131
- K(runch, 124
 - E(xamine and, 125

Index

p-System Tutorial

leaving the p-System, 132

L(ist Directory, 100

M(ake, 105

 recovering files, 127

M(ake boot, 136

MARKDUPDIR, 128

menu, 86

M(onitor, 132

M(onitor file

 creating, 133

 eX(ecuting, 133

N(ew, 120

options, 87

 displaying, 88

 Filer, 104

p-Code, 82

p-System

 leaving, 132

 starting, 85

P(refix, 103, 117

prefixed disk, *see* default disk

PRINTER:, 101, 117, *see also* PRINT utility

 address port, 138

 configuring, 138

 printer initialization string, 138

 printing file, *see* Transfer

promptline, 86

Q(uit, 105

REAL2.CODE, 145

REAL4.CODE, 145

RECOVER, 125

recovering files

 M(ake, 105, 127

 RECOVER, 125

reinitializing, *see* I(nitialize

REMIN:, 101

REMOUT:, 101

R(emove, 111

root volume, *see* system disk

S(ave, 119

screen, *see also* CONSOLE:

 displaying files on, 116

starting the p-System, 85

system disk, 102

 copying, 143

 listing, 112

SYSTEM files, 143

system tracks, *see* boot tracks

SYSTEM.CHARSET, 143

SYSTEM.FILER, 143

SYSTEM.INTERP, 143, 145

SYSTEM.MISCINFO, 143

SYSTEM.PASCAL, 143, 145

SYSTEM.WRK.TEXT, 119, *see also* work files

 removing, 120

SYSTEM:, 100

TEXT files, 99

 recovering, 126

T(ransfer, 113

 entire diskette, 115

 on the same disk, 114

 several files, 114

 single files, 113

 to the printer, 117

 to the screen, 116

TWO.BOOT, 146

Index

p-System Tutorial

Universal Medium, 141
 block size, 144
 U(ser Restart, 131
 UTIL, 135
 B(ack-up, 139
 C(onfigure, 137
 F(ormat, 92
 M(ake boot, 136

 V(ol, 112
 volume names, 100
 changing, 111
 CONSOLE:, 100
 creating, 95
 disk name, 101
 listing, 112
 PRINTER:, 101
 REMIN:, 101
 REMOUT:, 101
 SYSTEMR:, 100
 use of, 102
 volume numbers, 100
 volumes, 100
 default, *see* default volume
 on-line, 112
 system, *see* system disk

 W(hat, 119
 wild cards, 107
 \$, 107
 =, 107
 ?, 107
 work files, 118
 creating, 118
 Editor and, 119
 listing, 119
 removing, 119
 WRK files, 99

 Z(ero, 94
 ZERO.BOOT, 146

p-System Editor

/ (repeat factor), 157, 173
 < (direction indicator), 151
 <sub>, *see* substitute phrase
 <targ>, *see* target strings
 > (direction indicator), 151
 ^C, 169

 A(djust, 176
 auto-indent, 167, 182
 setting, 167, 193

 centering lines, 176
 command character
 M(argin, 187
 setting, 194
 C(opy, 177
 setting markers, 196
 copy buffer, 169, 177
 copying from text files, 177
 copying text, 169
 cursor movement commands, 172
 cursor movement keys, 172

 D(elete, 161, 178
 copy buffer, 169
 deleting text, 178, 199
 direction indicator, 150, 156, 173

 Editor
 creating files, 158
 errors when saving file, 189
 leaving, 157, 171
 options, 150, 173, 174
 starting, 149, 170
 window, 151
 E(nvironment, *see* S(et
 ESC, 169
 etx, *see* ^C

Index

p-System Editor

- exchanging text, 198
- E(xit, 158, 188
- filling, 167, 182
 - setting, 167, 193
- F(ind, 162, 179
 - marking characters, 163
 - repeat factors, 162
 - searching for a previous phrase, 165
- finding text, 179
- global direction, 150, *see also* direction indicator
- I(nsert, 154, 160, 166, 181
 - auto-indent, 166, 182
 - copy buffer, 170
 - filling, 166, 182
- J(ump, 184
 - setting markers, 196
- K(olumn, 184
- leaving the Editor, 157, 171, 188
- L(it, *see* literal mode
- literal mode, 164
 - setting, 195
- M(argin, 168, 185
 - command character and, 187
 - copy buffer, 170
- margins, 166
 - A(djust, 176
 - I(nsert, 166, 183
 - M(argin, 168, 185
 - setting, 193
- markers, 196
- marking characters, *see* F(ind or R(eplace
 - menu, 150
 - moving columns of text, 184
 - moving text, 169
- options, 150, 173, 174
- P(age, 187
- Q(uit, 158, 187
 - E(xit, 158, 188
 - R(eturn, 158, 188
 - U(pdate, 158, 187
 - W(rite, 158, 188
- reformatting paragraphs, 185
- repeat factors, 157, 173
 - /, 157, 173
 - F(ind, 162
 - R(eplace, 162
- R(eplace, 162, 190
 - marking characters, 163
 - repeat factors, 162, 190
 - searching for a previous phrase, 165
 - verifying the phrase, 164, 191
- replacing text, 190
- R(eturn, 158, 188
- saving text files, 188
- S(et, 192
 - E(nvironment, 192
 - M(arker, 196
- starting the Editor, 149, 170
- substitute phrase, 163
- SYSTEM.WRK.TEXT, 188
- tabstops, 194
- target phrase, 163

Index

p-System Editor

TEXT files
 editing, 152
T(ok, *see* token mode
token mode, 164
 setting, 195

U(pdate, 158, 187

V(erify, 198
V(fy, 164, 191

window, 151
work file, 171
workspace, 158
W(rite, 158, 188

X(change, 198

Z(ap, 199
 copy buffer, 170

p-System PRINT

A(dvance, 221

B(ottom margin, 207

choosing the output, 220
COMMAND, 216
command character, 212
 setting, 213, 216
command line, 212
 COMMAND, 216
 END, 217
 ESCAPE, 218
 HEADING, 215
 INCLUDE, 214
 PAGE, 215

Date, 209
defaults
 setting, 222
D(ouble spacing, 210

END, 217
ESCAPE, 218
escape flag
 changing, 210
 setting, 218
escape sequences, 208
 Date, 209
 File, 209
 format, 209
 in your text file, 212
 Page, 209
eX(ecuting script files, 223

File, 208
F(irst page, 210
formfeeds, 221

Index

p-System PRINT

G(o, 221

H(eader line, 208, *see also* HEADING

 Date, 209

 File, 208

 format, 209

 Page, 208

HEADING, 215

INCLUDE, 214

I(nput, 219

leaving PRINT, 222

line

 Header, *see* Header line

 numbering, *see* Numbering lines

 spacing, *see* Double spacing

M(ake a script file, 222

margins

 printing, 206

N(umbering lines, 211

options, 203

O(utput, 220

Page, 208, 215

page breaks, 215

page layout, 206, 207

page numbering, 210

P(age size, 206

PRINT

 leaving, 222

 options, 203

 starting, 203

printing

 several files, 214

 starting, 221

script file, 222

setting the defaults, 222

single sheets

 loading, 220

starting PRINT, 203

starting printing, 221

S(top, 220

T(op margin, 206

U(se formfeed, 221

NOTES

NOTES

NOTES



OSBORNE™
COMPUTER CORPORATION