

APPLE WRITER][
OPERATING MANUAL

Written by: Paul Lutus
and Liane Finstad

In conjunction with Apple Computer Inc.

NOTICE

Apple Computer, Inc. reserves the right to make improvements in the product described in this manual at any time and without notice.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

APPLE COMPUTER, INC. AND PAUL LUTUS MAKE NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. APPLE COMPUTER SOFTWARE IS SOLD OR LICENSED "AS IS". THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT APPLE COMPUTER, INC., OR PAUL LUTUS, THEIR DISTRIBUTORS, OR THEIR RETAILERS) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL APPLE COMPUTER, INC., OR PAUL LUTUS BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF THEY HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

This manual is copyrighted. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Apple Computer, INC.

1981 by APPLE COMPUTER, INC.
10260 Bandley Drive
Cupertino, California 95014
(408) 996-1010

The word APPLE and the Apple logo are registered trademarks of APPLE COMPUTER, INC.

Special Delivery Software is a trademark of APPLE COMPUTER, INC.

TABLE OF CONTENTS

<u>INTRODUCTION</u>	1
<u>CHAPTER 1</u> <u>OPERATION OF THE TEXT EDITOR</u>	2
GETTING STARTED.....	2
What You Need.....	2
THE EDITOR MENU.....	3
Getting Apple Writer "Up and Running".....	3
CTRL-Characters.....	4
Returning to the Editor Menu.....	4
THE TEXT EDITOR DISPLAY.....	5
The Data Line.....	5
Removing the Data Line.....	5
TEXT ENTRY.....	6
The Cursor.....	6
Entering Text.....	6
Capitalizing Using the ESC Key.....	6
Memory.....	7
MOVING THE CURSOR.....	7
The ESC Key.....	7
Data Line "Pos:".....	8
CTRL-B ([B]eginning).....	8
CTRL-E ([E]nd).....	8
Display Scrolling.....	8
Keystroke Summary for Cursor Movement.....	9
CORRECTING TYPING ERRORS.....	10
Deleting Characters - LEFT ARROW Key <--.....	10
Inserting Text.....	10
Retrieving Characters - RIGHT ARROW Key -->.....	10
Summary.....	10
CTRL-K ALPHA LOCK.....	11
WORD WRAP-AROUND.....	11
CTRL-Z.....	11
FOR HELP.....	12
Returning to the Help Screens.....	13
THE ADDITIONAL FUNCTIONS MENU.....	14
CTRL-Q.....	14
CAPITALIZING USING THE SHIFT KEY.....	15
SAVING YOUR TEXT.....	16
File Names.....	16
Drive Number.....	16
CTRL-S [S]ave Contents of Memory to Diskette.....	17

TABLE OF CONTENTS

Data Line "File:".....	17
[S]ave := Using File Name on Data Line.....	18
[S]aving a Segment of Text.....	18
Delimiters.....	18
Markers.....	18
[S]ave - Add To A Disk File.....	19
Keystroke Summary of [S]ave Commands.....	20
 ERASING MEMORY CONTENTS.....	 21
[N]ew.....	21
 LOADING FILES INTO MEMORY.....	 21
CTRL- L [L]oad Contents of File into Memory.....	21
Drive Number.....	22
Data Line "File:".....	22
[L]oad :? Forgiven File Name.....	22
[L]oad a File Using =.....	23
[L]oad a Segment of a File.....	23
[L]oad a File Segment Exclusive of Markers.....	24
[L]oad All Occurrences of a Text Segment.....	24
Combining [L]oad Options.....	24
[L]oad :# Copying Text in Memory.....	24
[L] Used to Display the File on the Screen.....	25
Memory Overflow.....	25
Keystroke Summary of [L]oad Commands.....	26
 CTRL-0 DOS COMMANDS MENU.....	 27
CTRL-0.....	27
Selection #1 Catalog.....	27
"Sectors" Defined.....	28
Checking File Length.....	28
Selection 2. Rename File.....	28
Selection 3. Lock File.....	29
Selection 4. Unlock File.....	29
Selection 5. Delete File.....	29
Selection 6. Initialize diskette.....	30
 TABULATION.....	 30
CTRL-L.....	30
Data Line "Tab:".....	30
CTRL-T [T]ab(Set/Clear/Purge).....	30
 SAVING AND RETRIEVING TAB SETTINGS.....	 31
CTRL-Q2 Save Tab File.....	31
CTRL-Q1 Load Tab File.....	31
Tab File Name Prefix.....	31
Changing Default Tab Settings.....	32
 "SEEING" RETURNS.....	 32
CTRL-Q6 Toggle Carriage Return Display.....	32
 UNDERLINING.....	 33
Reverse Slash.....	33

TABLE OF CONTENTS

MORE ON TEXT DELETION AND RETRIEVAL.....	34
CTRL-D [D]irection of Data Line Arrow.....	34
CTRL-W [W]ord Deletion/Retrieval.....	34
CTRL-X Paragraph Deletion/Retrieval.....	35
CTRL-R [R]eplace.....	35
AUTOMATED TEXT SEARCH AND REPLACEMENT.....	36
[F]ind - Search Only.....	36
CTRL-F [F]ind Text String and Replace.....	37
[F]ind and Replace Automatically.....	38
[F]ind - Replace Something with Nothing.....	38
[F]ind := Repeat the Previous Search.....	38
[F]ind - An Example Application.....	39
Keystroke Summary of [F]ind Commands.....	40
USING [F]IND WITH SPECIAL DELIMITERS.....	41
Using Other Delimiters.....	41
Possible Delimiter Selections & Associated Special Symbols	42
CREATING YOUR OWN GLOSSARY.....	43
Defining Glossary Terms.....	43
Inserting [G]lossary Terms into Text.....	44
Saving a List of Glossary Definitions.....	44
Deleting Glossary Definitions from Memory.....	45
Keystroke Summary of [G]lossary Commands.....	46
SUMMARY OF THE DATA LINE DISPLAY.....	47
CTRL Q8 - DATA LINE TOGGLE.....	48
CONVERTING APPLE WRITER 1.1 FILES.....	48
SPECIAL FEATURES AND TECHNIQUES.....	49
CTRL-C Automatic [C]ase Change.....	49
CTRL-Y Split Screen Operation.....	50
CTRL-V CTRL-characters as Text Entries.....	51
Superscripts and Subscripts.....	51
Footnote Entry Procedure.....	52
SPECIAL CHARACTERS.....	53
PROTECTING YOURSELF AGAINST DISASTER.....	54
DISK ERROR MESSAGES.....	54
IN CASE OF TROUBLE.....	55
QUITTING APPLE WRITER][.....	56

TABLE OF CONTENTS

<u>CHAPTER 2</u>	<u>PRINTING THE TEXT FILE</u>	57
PAGE FORMAT COMMANDS.....		58
CTRL-P ? [P]rint/Program Commands-Display.....		58
Format Command Codes and Default Values.....		59
CHANGING FORMAT VALUES USING CTRL-P.....		61
[P]? - Using the Print/Program Commands Display as a Menu.		61
[P] Changing Format Values Without Using the Menu.....		62
TOP AND BOTTOM LINE ENTRY PROCEDURE.....		62
Printing Page Numbers.....		64
OPTIONS FOR JUSTIFYING YOUR TEXT.....		64
Left Justify [P]rint/Program :LJ.....		64
Center Justify [P]rint/Program :CJ.....		65
Right Justify [P]rint/Program :RJ.....		65
Fill Justify [P]rint/Program :FJ.....		65
TEXT-EMBEDDING PRINTING COMMANDS.....		66
Embedding Format Values.....		66
Indenting and Outdenting Paragraph Margins - .PM.....		66
Centering a Line.....		67
ACTUALLY PRINTING.....		68
[P] NP - New Print.....		68
Stopping the Printer.....		68
[P] CP - Continue Print.....		68
Printing Only Selected Pages of a Document - .EP{0 or 1}..		69
When You DON'T Want a Text Line Printed.....		69
Suppressing First Page "TL" and "BL" Entries.....		69
ADDITIONAL PRINTING COMMANDS.....		70
Form Feed - .FF.....		70
Form Feed Only If - .FF{number}.....		70
Automatic Prompting - .IN.....		71
SAVING AND LOADING [P]RINT/PROGRAM COMMAND VALUES.....		71
Permanently Changing Default [P]rint/Program Values.....		72
SUMMARY OF PRINTING PROCEDURE.....		73

TABLE OF CONTENTS

<u>CHAPTER 3</u> <u>WORD PROCESSING LANGUAGE</u>	74
KEY FEATURES.....	74
WPL COMMAND ORIGINS.....	75
HOW TO WRITE A WPL PROGRAM.....	75
LESSON 1. TYPICAL WPL PROGRAM.....	76
Lesson 1 Summary.....	78
LESSON 2. REPEATING PARTS OF THE PROGRAM.....	79
LESSON 3. USING NUMBERS IN THE PROGRAM.....	81
LESSON 4. USING NUMBERS TO RETRIEVE TEXT FILE SEGMENTS.....	82
LESSON 5. ERROR FLAGGING AND DETECTION.....	84
LESSON 6. WPL PROGRAM TERMINATION.....	85
SUMMARY OF WAYS TO END A WPL PROGRAM.....	86
LESSON 7. AUTOMATIC TABLE INDEXING.....	87
LESSON 8. PERSONALIZING A LETTER BEYOND THE ADDRESS.....	88
WPL SYNTAX ERRORS.....	89
ADVANCED WPL TECHNIQUES AND FEATURES.....	91
STRINGS IN WPL.....	91
WPL SUBROUTINES.....	93
SCREEN DISPLAY DURING WPL PROGRAMS.....	94
WPL PROGRAM COMMENTS.....	95
WPL PROGRAM LENGTH.....	95
CONTROLLING TEXT EDITOR DISPLAY FROM WPL PROGRAMS.....	96
WPL LIBRARY CREATION.....	96
WPL EXECUTION ERRORS.....	96
"WPL Error: Label not found --> xxxxxx.....	96
"WPL Error: 'RT' without 'SR'.....	97
"WPL Error: Program > 2048 chars.....	97
"WPL Error: More than 32 'SR'.....	97
"WPL Error: Footnote overflow.....	97

TABLE OF CONTENTS

WPL EXAMPLE PROGRAMS.....	98
Word Counter.....	98
Autoletter.....	99
File Mover.....	100
Automatic Replacement.....	101
Automatic Print.....	102
APPENDIX A WPL COMMANDS SUMMARY.....	103
APPENDIX B SPECIAL CHARACTERS GLOSSARY FILE.....	104
APPENDIX C LOADING THE DEFAULT TAB SETTINGS AND [P]RINT/PROGRAM VALUES.....	105
APPENDIX D USING APPLE WRITER][WITH GOODSPELL.....	106

INTRODUCTION

Welcome to the world of word processing! Apple Writer][is a computer program that lets you create, modify and print text. It also uses a simple but powerful computer language called WPL (Word Processing Language) to automate the process of text manipulation and document creation.

Using Apple Writer][you may:

- * Write letters, reports and books using a powerful text editor program.
- * Insert, delete, or change characters, words, lines, or whole chapters anywhere in your document, either manually or automatically.
- * Print the text with printing format adjusted to your specifications as printing is underway.
- * Command the printing of form letters or other repetitive documents, with names and addresses and other personalizing information automatically inserted as the letter is printed.
- * Create application programs to automatically carry out complex word processing tasks such as rearranging documents, finding and changing references within text, and creating reports out of diverse elements.
- * Translate text from typewriter shorthand to English phrases (and other languages) and back again. You may define the symbolic sets and the resultant text form.

This manual starts with the features of Apple Writer][that you will use most frequently, and moves from the simple toward the more complex options. Therefore it is a good idea to read it from the beginning, performing each example. The manual may then be used for reference by examining the Table of Contents and the Reference Card.

>>> GETTING STARTED

What You Need: To use Apple Writer II you need the diskette containing Apple Writer II, an Apple II computer with a minimum of 48K of memory, one or more disk drives, and a video monitor.

As an option, you may want to install a "Sup'R'Terminal" 80 column display card (M & R Enterprises, Sunnyvale, California). This display card must be operated in Apple peripheral slot 3. Ask your dealer for information.

It is advisable to also purchase an additional monitor cable to allow easy switching between 40 column and 80 column operation. It is also recommended that the "Shift-key Modification" be made. This modification permits the entry of upper case letters by using the shift key just like a typewriter. References to the "shift-key modification" in this manual refer to this change. Ask your dealer to make this change for you.

For reference, you should have on hand your Apple II DOS Manual.

To use the printer program to print a document, you need an Apple compatible printer (and its interface card if appropriate).

You will also need to have on hand extra diskettes which have been initialized (this procedure is explained in your Apple II DOS Manual and in this manual under the "CTRL-O DOS Commands Menu" section) for file storage so that you may save your documents and make "back-up" copies of your work to protect yourself against "accidents".

OPERATION OF THE TEXT EDITOR

CTRL-Characters: The list of "Frequently Used Control Characters" that you see is a list of the Apple Writer][commands that you will be using most often. The bracketed letter which appears before each selection is a "shorthand" notation for a feature which is obtained by holding down the CTRL key (located on the left edge of the keyboard) while you press the appropriate letter. CTRL-characters are the way you get the attention of the Apple Writer][program. They tell the program that you want to use one of its functions. To enter a CTRL-character command, press the CTRL key first and continue to hold it down while you press the desired character. In these instructions, CTRL-character commands are preceded by "CTRL-" or appear in brackets "[L]" as they do in the Editor Menu.

NOTE: THE CHARACTER IN A CTRL-CHARACTER COMMAND MAY BE ENTERED IN UPPER OR lower CASE.

Any of the CTRL-character commands which appear on the Menu may be selected by pressing the CTRL key and the appropriate letter. For example, if you press CTRL-L ([L] Load a file), the following message appears at the bottom of your screen:

[L]oad :

You would type the name of the file you wished to load into memory and press RETURN. A full explanation of the operation of these commands is given in the manual. To find the one you want, check the Table of Contents. If you make a mistake and press the wrong CTRL-character command, simply press return to "undo" the command.

NOTE: The CTRL-character commands given in the menu may also be used while you are entering or modifying text by pressing the appropriate CTRL-character. It is not necessary to return to the Editor Menu in order to enter Apple Writer][commands.

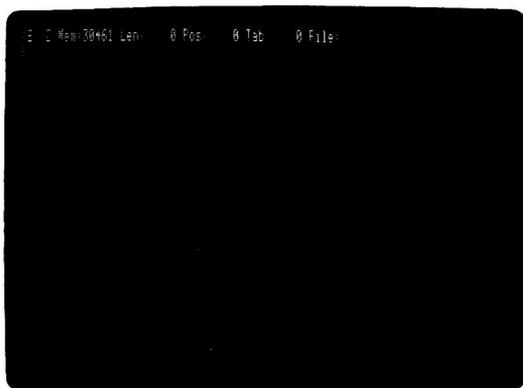
To begin entering text for a new document, press RETURN. The editor program is now ready to accept text entries.

Returning to the Editor Menu: You may return to the Editor Menu display at any time by pressing ESC ESC ?

OPERATION OF THE TEXT EDITOR

>>> THE TEXT EDITOR DISPLAY

The Data Line: When you pressed RETURN to leave the Editor Menu, you are in the text entry mode. You will see some numbers and other information on the top line of your screen as shown below.



This is called the "Data Line". Each of the items on the Data Line will be fully explained as it is used.

Removing the Data Line: If you are distracted by the Data Line, you may remove it by using the Additional Functions Menu, selection 8 ("Toggle Data Line Display"). See the "Additional Functions Menu" section of the manual for a list of these editor options. If you press [Q] and enter 8 in response to the prompt, the Data Line will be removed from your screen. To display the Data Line again, press [Q]8 again. This command is called a "toggle" command which means that the command is switched on or off using the same keystroke.

>>> TEXT ENTRY

The Cursor: Immediately below the Data Line you will see a flashing white rectangle called the "cursor". The cursor is like the position of the carriage on a typewriter. It leads you through the text, indicating the position at which any text entries, insertions or deletions will be made. Characters, such as @ or ©, may be displayed within the cursor to let you know which special editor features you have selected.

Entering Text: To enter text, use your Apple][keyboard just like a typewriter. As you enter text, the words appear on the screen, and the cursor moves to the right. When the cursor gets beyond the right screen edge, it will reappear at the left edge, one line down. It is NOT necessary (nor desirable) to press RETURN to start a new line. THE RETURN KEY SHOULD BE PRESSED ONLY AT THE END OF A PARAGRAPH OR FOR BLANK LINES BETWEEN PARAGRAPHS.

Capitalizing Using the ESC Key: When you first boot Apple Writer][an "E" is displayed in the second position on the Data Line. This "E" means that you must use the ESC key (located on the left edge of the keyboard) to enter capital letters. Here's how:

1. Press ESC once. The cursor will contain a "@" to let you know that the next alpha character entered will be capitalized.
2. Now type the desired letter. The capital letter will appear on your screen and the cursor will again be blank.

You must press ESC once for each letter you wish to capitalize. If you wish to enter several words or phrases in upper case, Apple Writer][offers an editor feature which allows you to do this without pressing ESC for each letter. See the "CoNTRoL-K Alpha Lock" section of the manual.

NOTE: If you do not have an 80-column board and the display on your screen is upper case, capital letters will appear as inverse (black on white) characters.

NOTE: The SHIFT key must be used to enter any special, non-alpha characters such as \$, %, * and so forth.

OPERATION OF THE TEXT EDITOR

Memory: Each character that you type appears on the screen and is saved for you in the computer's memory. In this manual, the phrase "text in memory" refers to the typing which appears on your screen. You can tell how many characters you have entered into memory by checking the number which appears next to "Len:" (length) on the Data Line. This number gives the count of how many characters are presently in memory. The number which appears next to "Mem:" (memory) gives the number of characters which may be entered into memory. As you enter text, the "Mem:" number will decrease, and the "Len:" number will increase.

If "Mem:" is zero, you will not be able to make any more entries. See the "Saving Your Text" section in the manual.

>>> MOVING THE CURSOR

The ESC Key: The cursor can be moved anywhere within the text on the screen (but not beyond the last character entered) without affecting the text in memory. To change the cursor from the text entry mode to move mode, press the ESC key (located on the left edge of the keyboard) TWICE. The cursor will now contain a "@" character to indicate that it is in move mode. The following keys can now be used to move the cursor:

<u>Press</u>	<u>Cursor Moves</u>
I	up one line
J	left one character
K	right one character
M	down one line
E	up 12 lines
S	left 24 characters or to the next space
D	right 24 characters or to the next space
X	down 12 lines

NOTE: If you have made the "Shift-key Modification", you only need to press the ESC key ONCE to change to the cursor move mode.

To move the cursor faster, hold down the REPT key (located on the upper right of the keyboard) and the desired cursor move key at the same time.

OPERATION OF THE TEXT EDITOR

Note that the position of the letters E, S, D and X on the left side of the keyboard corresponds to the position of the letters, I, J, K, and M on the right. You may wish to make labels for the keys as a memory aid, or imagine them like this:

	E		I	
S		D		J K
	X			M

To return to text entry mode, press any character except the cursor move keys (I,J,K,M,E,S,D,X) or ?. The cursor will again be blank to indicate that you are in text entry mode.

Data Line "Pos:": As you move the cursor, the number next to "Pos:" on the Data Line will change. This number represents the character number of the cursor's present position from the beginning of the file. For example, if you have entered 50 characters of text in memory and moved the cursor to the middle of your entry, "Pos:" would read "25" and "Len:" would read "50".

CTRL-B ([B]eginning): A quick way to move the cursor to the beginning of the text in memory is to use [B]. The [B] command also sets the first character on the Data Line to >. This character is called the "direction arrow". The direction arrow is used in conjunction with some special editor features, and its significance will be explained when these special features are introduced.

CTRL-E ([E]nd): To move the cursor to the end of the text in memory, press [E]. The [E] command also sets the direction arrow on the Data Line to <.

Display Scrolling: If much text has been typed in, it may extend beyond the top of the screen. To see this text, simply move the cursor toward the top of the screen. The text will scroll downward to keep the cursor in view. In the same way, if you move the cursor toward the bottom of the screen, the text will scroll up.

OPERATION OF THE TEXT EDITOR

KEYSTROKE SUMMARY FOR CURSOR MOVEMENT

Press ESC twice, then press:

I	Moves cursor up one line
J	Moves cursor left one character
K	Moves cursor right one character
M	Moves cursor down one line
E	Moves cursor up 12 lines
S	Moves cursor left 24 characters or to the next space
D	Moves cursor right 24 characters or to the next space
X	Moves cursor down 12 lines

Press:

[B]	Moves cursor to beginning of text in memory. Sets Data Line direction arrow to >.
[E]	Moves cursor to end of text in memory. Sets Data Line direction arrow to <.

>>> CORRECTING TYPING ERRORS

Deleting Characters - LEFT ARROW Key <--: If you make a mistake when you are typing text, you may delete unwanted characters by first moving the cursor to the immediate RIGHT of the character you wish to delete. Then press the LEFT Arrow key. To delete a string of characters, hold down the REPT key and the left Arrow key at the same time.

Inserting Text: If you need to insert a word in the middle of a sentence, or a sentence in the middle of a paragraph, or a paragraph in a document, position the cursor where you want to insert the text and start typing. Any text in "front" (to the right) of the cursor will be pushed across the line and down the "page" of your screen to make room for your insertion.

Retrieving Characters - RIGHT ARROW Key -->: Characters deleted using the left Arrow key are saved for you in what is called a "storage buffer". The buffer holds 128 characters. These deleted characters may be reinserted at the position of the cursor by pressing the right Arrow key. Characters retrieved in this fashion appear to the left of the cursor just as though you had typed them. The first character which appears will be the LAST one which you deleted. This is a useful way to move a few words from one portion of text to another. Just delete the characters using the LEFT Arrow. Reposition the cursor to where you want to insert the text and press the RIGHT Arrow.

Summary: Characters are deleted from right to left using <--. These characters are saved and may be reinserted (last character deleted appears first) in the text by pressing -->. New text may be inserted by entering the desired characters at the position of the cursor.

OPERATION OF THE TEXT EDITOR

>>> CTRL-K ALPHA LOCK

If you wish to enter several words or phrases in upper case, Apple Writer][offers an editor feature which allows you to do this without pressing ESC (or the SHIFT-key if you made the modification) for each capital letter. To enable the alpha lock feature, press [K]. A "U" will be displayed on the Data Line in place of the the "E" to let you know that you have selected this editor feature. When [K] is enabled, only letters which you type will appear in UPPER CASE. To enter the special characters which appear in the "upper" position on the keyboard you will still need to use the SHIFT key.

To disable the alpha lock feature, press [K] again. The "U" on the Data Line will be replaced by "E".

>>> WORD WRAP-AROUND

The text editor display automatically "wraps" words around the screen. This means that if a word is too long to fit on a line, it is moved down to the next line.

CTRL-Z: You may want to disable the word wrap-around feature for aligning columns of text, or to display carriage returns (refer to the "Seeing Returns" section of the manual). To disable the word wrap-around feature, press [Z]. The "Z" on the Data Line will disappear to remind you that word wrap-around has been disabled. To regain word wrap-around press [Z] again, and the "Z" will reappear on the Data Line.

NOTE: When formatting text in columns, it is a good idea to disable the word wrap-around feature by pressing [Z].

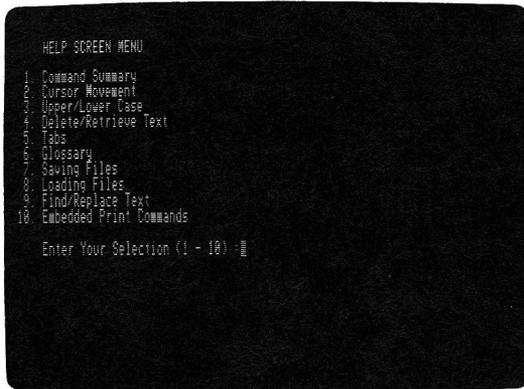
OPERATION OF THE TEXT EDITOR

>>> FOR HELP

At the end of the "Frequently Used Control Characters" list that you see when you first boot Apple Writer][there is a note which says:

(for help, enter "?")

If you enter a question mark (don't press RETURN) you will see:



NOTE: IN ORDER TO VIEW THE HELP SCREENS, YOUR APPLE WRITER][MASTER DISKETTE MUST BE IN DRIVE 1.

To see one of these screens, enter the desired number and press RETURN. You will see a brief summary of the entries required to use a particular editor function.

To exit the Help Screen Menu, press RETURN. At the bottom of each Help Screen you will see the message:

Press "C" to continue or "E" to exit, and then press RETURN

If you wish to see another Help Screen, press "C" and then press RETURN. If you wish to return to the editor program, press "E" and then press RETURN.

OPERATION OF THE TEXT EDITOR

Returning to the Help Screens: You may look at the Help Screens at any time during the operation of Apple Writer][by following this sequence of entries:

1. Press ESC ESC ?. Your screen will display the Editor Menu.
2. Now enter ? (Don't press RETURN)

The Help Screens Menu will appear on your screen and you may enter the selection of your choice. The Reference Card at the back of the manual also gives a list of the editor commands.

You may also recall the Help Screens without returning to the Editor Menu by the following entries:

[P] DO HELP80 (if you have an 80-column card)

or

[P] DO HELP40 (if you don't have an 80-column card)

These are WPL commands which are explained in the WPL section of the manual.

>>> THE ADDITIONAL FUNCTIONS MENU

CTRL-Q: One of the frequently used control-characters listed on the Editor Menu is "[Q] Additional Functions Menu". If you press [Q], your screen will display:

```

ADDITIONAL FUNCTIONS MENU
1. Load Tab File
2. Save Tab File
3. Load Print/Program Value File
4. Save Print/Program Value File
5. Load [C] Glossary File
6. Toggle Carriage Return Display
7. Toggle Shift Key Beep (Esc/SHIFT)
8. Toggle Beta Line Display
9. Convert Apple Writer 1.1 Files
0. Quit Apple Writer

Enter your selection (1-0) :

```

These options may be chosen by entering the appropriate number. You may leave the Additional Functions Menu without entering a selection by pressing RETURN. The functions of these options will be explained as each feature is introduced in the manual.

>>> CAPITALIZING USING THE SHIFT KEY

NOTE: THIS EDITOR FEATURE WILL OPERATE ONLY IF THE SHIFT-KEY MODIFICATION HAS BEEN MADE.

If you have made the shift-key modification, you may follow this procedure to enter capital letters using the SHIFT key.

1. Press [Q]. The Additional Functions Menu will be displayed on your screen.
2. Enter 7 in response to the "Enter Your Selection (number) :" prompt. This is the "Toggle Shift Key Option (ESC/SHIFT)" menu option.

The Data Line will now display an "S" (S = Shift) in the second position and you may use the SHIFT key to enter upper case letters. See the "Permanently changing Default [P]rint/Program Values" section of the manual to permanently change the default.

To return to the ESC method of capitalizing, press [Q]7 again. The "S" on the Data Line will be replaced by an "E" to let you know that you must use the ESC key for entering capitals.

NOTE: There are a few special characters which can only be obtained using the ESC method of capitalization. See the "Special Characters" section of the manual.

>>> SAVING YOUR TEXT

The text on the screen is in the computer's memory. If the power were suddenly shut off, this text would be lost. To save text for future use and to protect yourself from "accidents", it must be recorded on diskette. In fact, it is good practice to make a habit of "saving" your text on diskette every 15 minutes or so.

File Names: The text on the screen (and in the computer's memory) is also called a "file". A file can be as short as one character or as long as the computer has space for in its memory. In order to save the file in memory on a diskette you have to give it a name (like putting a label on a folder). The name you give your file must conform to the format and rules given in The DOS Manual. Here is a short summary:

1. A file name can contain up to 30 characters.
2. A file name must begin with a letter.
3. A file name cannot contain a comma.
4. File names can be entered in upper or lower case. Apple Writer]L will automatically convert lower case file name entries to upper case.

Here are a few legal file names:

```
MIKE 2 JULY 80
sortprogram
LETTER TO SUE
```

Here are some names than will NOT work, and reasons why:

```
1 TO 10           (begins with a number)
PROGRAM,STEVE    (contains a comma)
```

Drive Number: File names entered when using [S]ave may be suffixed by a slot and drive number. You may use any appropriate slot and drive numbers following the file name as explained in The DOS Manual. If you only have one drive, you can ignore all references to drive number in this manual.

CTRL-S [S]ave Contents of Memory to Diskette: To save the file in the computer's memory to a diskette, press [S]. The Apple Writer program will respond by displaying at the bottom of your screen:

[S]ave :

The program is waiting for you to tell it what to call the file. If you want to call your file "TEST" and the diskette you want to store the file on is in drive 1, you would type:

TEST,D1

and press RETURN. The red indicator light on the disk drive will come on, the drive will whirl and the file will be saved.

If your diskette were in drive 2, you would type:

TEST,D2 and press RETURN

CAUTION!! If you enter a file name which already exists on the diskette, [S]ave will REPLACE whatever was previously stored under that file name with memory contents. To protect your important files from accidental replacement or erasure see the "CTRL-0 DOS Commands Menu" section of the manual for a method to "lock" your files.

Data Line "File:" The Data Line will contain, opposite "File:", the file name used for any [S]ave operation.

NOTE: Apple Writer][does not distinguish between upper and lower case letters in a file name. You may [S]ave a file using upper case and retrieve the same file using lower case and vice versa.

[S]ave := Using File Name on Data Line: If you are working on a fairly lengthy document and are saving your text regularly as suggested above, you may [S]ave your file under the name shown on the Data Line by entering [S]=. The contents previously saved under that file name are replaced by the current memory contents.

[S]aving a Segment of Text: If you don't want to save the entire file in memory (if you want to move large blocks of text, for example) you may save a portion of your text as follows:

Move the cursor to the beginning of the text segment you wish to save. Then enter:

```
[S]ave :FILENAME/marker/
```

Delimiters: The slashes used in this example are called "delimiters". They separate the file name from the characters you entered to tell the editor where the text segment you were saving ended.

Markers: The set of characters enclosed by the delimiters is called a "marker". When you use this feature of [S]ave, the editor searches the text forward from the position of the cursor until it finds a set of characters which match the ones enclosed by the delimiters. It is entirely possible (and in the nature of things very probable) that the text segment you wish to save will end with a set of characters which occur more than once. In such a case it is necessary to enter a special character (such as an asterisk or percent sign) at the end of your text segment to serve as a unique marker.

CAUTION: DO NOT USE THE / AS A MARKER OR AS PART OF THE FILE NAME. THE EDITOR PROGRAM WILL READ IT AS A DELIMITER.

Suppose you wanted to save only the sentence:

Winter reaps what Summer sows.

1. Move the cursor to the immediate left of Winter (at the beginning of the text segment to be saved).
2. Press [S]. The editor program responds with "[S]ave :"

3. You type: TEST2/sows./ and press RETURN

The sentence "Winter reaps what Summer sows." will be saved under the file name "TEST2".

IF YOU HAD PREVIOUSLY SAVED TEXT UNDER THE FILE NAME "TEST2", IT WOULD NOW BE REPLACED BY THE EXAMPLE SENTENCE.

[S]ave - Add To A Disk File: If you want to ADD a text segment to the end of an existing disk file, move the cursor to the beginning of the segment and press [S]. The editor program responds with:

[S]ave :

You type: FILENAME,D2/marker/+

The "+" sign tells the editor program to ADD the text segment to the end of the disk file instead of replacing it.

You may also add the entire contents of the file in memory to a disk file by entering:

[S]ave :FILENAME,D2+

The "+" is the last character entered.

CAUTION: It is possible to create disk files with the [S]ave + option which are too large to be retrieved into memory. To find out how large your files are, see the "DOS Commands Menu" section in the manual.

OPERATION OF THE TEXT EDITOR

KEYSTROKE SUMMARY OF [S]ave COMMANDS

- [S] FILENAME,D# Saves entire contents of memory to the diskette in drive #. REPLACES any text previously stored under the file name used.
- [S] ? Displays CATALOG of file names stored on diskette.
- [S] = Saves entire contents of memory under file name shown on Data Line. Replaces any text previously stored under the file name used.
- [S] FILENAME,D2/marker/ Saves text segment from position of cursor up to, and including, marker. Replaces any text previously stored under the file name used.
- [S] FILENAME,D2+ ADDS text file in memory to the end of a diskette file.

NOTE:

All [S]ave commands must be followed by a RETURN.

To abort the [S]ave command (if you haven't entered any characters after pressing [S]), press RETURN. If you have entered characters, backspace using the LEFT arrow to erase all of the characters entered, and then press RETURN.

>>> ERASING MEMORY CONTENTS

[N]ew To clear the display on your screen (erase memory contents) press [N]. At the bottom of the screen you will see the message:

[N]ew (Erase Memory) Yes/No ?

You now have the opportunity to change your mind. If you still want to erase memory contents, type Y (or y) and RETURN. Entering any other character and pressing RETURN (or just pressing RETURN) will abort the [N]ew command and the memory contents will be retained.

NOTE: Make sure that you have saved your document and that you erase memory before beginning work on a new document.

>>> LOADING FILES INTO MEMORY

CAUTION!! The [L]oad command DOES NOT ERASE current memory contents. [L]oad INSERTS a file into memory at the position of the cursor. If you want memory to contain only the file you are loading, make sure you erase memory ([N]ew) before using [L].

CTRL- L [L]oad Contents of File into Memory: To copy a file from a diskette into memory, press [L]. At the bottom of your screen you will see:

[L]oad :

You respond by typing the name of the file and press RETURN. Your entry might look like this:

[L]oad : test,d1

Whatever text had previously been saved under the file name TEST would be placed in memory AT THE POSITION OF THE CURSOR.

NOTE: Since the [L]oad command does not erase present memory contents, you can insert text from diskette files into any position you wish in memory.

OPERATION OF THE TEXT EDITOR

If the file you wish to load is on a diskette in drive 2, you would enter:

```
[L]oad : TEST,D2
```

Drive Number: File names entered when using [L]oad may be suffixed by a slot and drive number. You may use any appropriate slot and drive numbers following the file name as explained in The DOS Manual.

NOTE: If you attempt to load a file which does not exist (if you made a typing error in the file name, for instance) your Apple will buzz and the message

```
DOS Error :File not found      (Press RETURN)
```

will appear at the bottom of the screen. Don't panic. Just follow the directions and press RETURN and you will see the Editor Menu which first appears on your screen when you "turn on" Apple Writer][. Press RETURN again and you will be back at the beginning of the file in memory. You may then execute the [L]oad command again followed by the correct file name. See "[L]oad :?" below for instructions on how to see a list of the files stored on diskette.

Data Line "File:" When you [L]oad a file after clearing memory, the file name used will appear on the Data Line. Otherwise, the file name appearing on the Data Line is not changed during [L]oad operations.

[L]oad :? Forgiven File Name: If you can't remember the name of the file you want to load or just want to see a list of the files stored on a particular diskette, Press [L]? and RETURN. To see the files stored on the diskette in drive 2, enter

```
[L]oad :?,D2
```

Your screen will display a list of the files stored on the diskette. If you have a number of files stored, you will need to press RETURN several times to see the full list. The prompt "[L]oad :" will be displayed beneath the list of file names. You may now enter the name of the file you wish to load into memory, or you may abort the command by pressing RETURN. If the [L]oad command is aborted at this point, you will be returned to the file in memory at the original position of the cursor.

NOTE: For a preferable method of obtaining a list of files stored on diskette, see the "DOS Commands Menu" section of the manual.

[L]oad a File Using = : If the file name displayed on the Data Line is the one you want to load, enter = after pressing [L] and then press RETURN.

[L]oad a Segment of a File: The procedure for loading part of a file into memory is very similar to [S]aving a file segment. You specify the file name and the beginning and ending marker for the portion of text to load. Suppose that you have saved a file named TEST consisting only of the sentence "Winter reaps what Summer sows." To load the words "reaps what Summer" enter:

```
[L]oad :TEST/reaps/Summer/
```

Now suppose you want to load a segment of the file beginning with the word "reaps". You would enter just the beginning marker:

```
[L]oad :TEST/reaps/
```

This entry would cause everything from the word "reaps" to the end of the file to be loaded into memory at the position of the cursor. ("reaps what Summer sows.")

If you want to load all the text from the start of the file and ending with the word "what", you would enter just the ending marker:

```
[L]oad :TEST//what/
```

By not entering any characters between the first two delimiters (//), you have instructed the editor to load everything from the start of the file TEST to the word "what", inclusive.

[L]oad a File Segment Exclusive of Markers: It is also possible to load file segments exclusive of the identifying markers. The entry:

```
[L]oad :TEST/reaps/sows./N
```

will cause the phrase " what Summer " to be loaded into memory. The N after the third "/" instructs the editor to load into memory everything between, but NOT including "reaps" and "sows."

There are some additional special features available when using [L]oad.

[L]oad All Occurrences of a Text Segment: Suppose, for example, that you have a disk file of names called NAMES, and that each name ends with a period. To load ALL the names that include the name "John" enter:

```
[L]oad :NAMES,D2/John./A
```

The A after the third / means load ALL the text segments which begin with "John" and end with a period.

Combining [L]oad Options: The "A" and "N" [L]oad options may also be used together. For instance, the entry:

```
[L]oad :NAMES,D2/John./AN
```

will load into memory from the disk file "NAMES" ALL cases of text segments beginning with "John" and ending with ".", but NOT the markers "John" and ".".

[L]oad :# Copying Text in Memory: You may also load from the memory itself by using the character "#" instead of a file name. The entry:

```
[L]oad :#/marker1/marker2/
```

would copy into memory at the position of the cursor, all text from "marker1" to "marker2" inclusive. This option is used to copy sections of text in one position of the file in memory to another.

[L] Used to Display the File on the Screen: It is also possible to display a disk file on the screen without actually loading it into memory. Enter the reverse slash as the last character in the [L]oad command. To enter a reverse slash without the shift-key modification, press ESC and then press the SHIFT, CTRL, AND P keys simultaneously. If you have made the shift-key modification, the reverse slash is obtained by pressing SHIFT-CTRL-P. For instance, the entry:

```
[L]oad :NAMES,D2/John./\
```

would cause the screen to display all the text beginning with "John" and ending with "." from the disk file NAMES. While screen printing is in process, it may be stopped temporarily by holding down the [S] key. To resume printing, press [S] again.

This special feature of [L]oad makes it possible to review the contents of a file without inserting it into the memory and without disturbing the current file in memory.

NOTE: See the "Special Characters" section of the manual for a summary of the keystrokes necessary to obtain special characters.

Memory Overflow: It is possible to fill memory to overflowing using the [L]oad feature. If this happens, the loading operation will cease, a buzzer will sound and the Data Line will show:

```
Mem:      0
```

This means that the text file area is completely full. The current memory contents may be saved in this case, but it will be necessary to delete some of the text in memory before adding any more.

NOTE: To check the length of disk files, see the "DOS Commands Menu" section of the manual.

KEYSTROKE SUMMARY OF [L]LOAD COMMANDS

- [L] FILENAME,D# Loads entire contents of FILENAME into memory from diskette in drive #.
- [L] ?,D# Displays catalog of file names stored on diskette in drive #.
- [L] = Loads file name displayed on Data Line into memory.
- [L] FILENAME,D#/marker1/marker2/ Loads text segment beginning with marker1 and ending with marker2, inclusive.
- [L] FILENAME,D#/marker/ Loads all text from file, beginning with marker.
- [L] FILENAME,D#//marker/ Loads all text from the start of the file and ending with marker.
- [L] FILENAME,D#/marker1/marker2/A Loads all cases of text segment from file including markers.
- [L] FILENAME,D#/marker1/marker2/N Loads first case of text segment exclusive of markers.
- [L] #/marker1/marker2/ Copies text segment from file in memory to present cursor position.
- [L] FILENAME,D#\ Displays file on screen without loading it into memory. Display may be halted and restarted using [S].

NOTE:

All [L]oad commands must be followed by a RETURN.

All [L]oad commands will INSERT the file into memory at the position of the cursor (except the [L]\ option). [L]oad DOES NOT ERASE current memory contents.

To abort a [L]oad command, use the left arrow key to backspace over any entries and press RETURN.

OPERATION OF THE TEXT EDITOR

>>> CTRL-0 DOS COMMANDS MENU

CTRL-0: If you press [0] (use the letter "o", not zero), your screen will display:

```
DOS COMMANDS
1 Catalog
2 Rename File
3 Lock File
4 Unlock File
5 Delete File
6 Initialize diskette

Enter your selection (1 - 6) :
```

To use any of the DOS Commands options (explained below), enter the number of your selection. You will be prompted by the editor for further information. If you decide not to use any of the options, press RETURN without making any entries. You will be returned to the file in memory at the original position of the cursor.

Selection #1 Catalog: To see a list of the files stored on a diskette, enter 1 after the "Enter Selection" prompt. Your screen will display another prompt:

Enter Slot,Drive (Example S6,D1) :

Enter the slot and drive number of the drive containing the diskette whose file catalog you wish to see and press RETURN. (For example, you may enter S6,D2 or D2). Your screen will display a list of the files stored on the diskette. If you have a large number of files stored, you may need to press RETURN again to see the entire list. The message:

(press RETURN)

appears at the bottom of your screen. When you press RETURN you will be returned to the DOS Commands Menu. You may then enter another selection or press RETURN to get back to the file in memory.

"Sectors" Defined: The numbers to the left of the file names in the diskette file catalog give the amount of storage space each file occupies. This sector number multiplied by 256 gives the maximum number of characters contained in each file.

Checking File Length: To make sure you have enough computer memory remaining to [L]oad a file, multiply its sector number by 256 and compare the result with the number which appears next to "Mem:" on the Data Line. If the sector number multiplied by 256 is larger than "Mem:", there is probably not enough memory available to [L]oad the file.

Selection 2. Rename File: If you wish to change the file name of a file, enter 2 after the "Enter Selection" prompt. Your screen will display this prompt:

Enter File Name :

Enter the file name which now appears in the catalog, a comma, then the desired new file name and press RETURN. For example,

Enter File Name : TEST,NEWTST

The file name will be changed, and you may make another selection from the DOS Commands Menu or press RETURN again to return to the text in memory.

You may abort this option by backspacing with the left arrow key over any entry made in response to the file name prompt and pressing RETURN.

NOTE: This command will not work if the file has been "locked". See "Selection 3" and "Selection 4" below.

Selection 3. Lock File: To protect an important file from accidental deletion or replacement, you may "lock" it. When a file has been locked, it may be [L]oaded into memory, but you will not be able to [S]ave using that file name, nor will you be able to rename or delete (explained below) that file name. To lock a file, enter 3 after the "Enter Selection" prompt. You will see the message:

Enter File Name :

Enter the name of the file you wish to lock and press RETURN. The file is now locked, and an asterisk will be displayed by that file name in the catalog. You may now make another selection from the DOS Commands Menu or press RETURN again to get back to the file in memory.

Selection 4. Unlock File: To remove a lock placed on a file under option 3, enter 4 after the "Enter Selection" prompt. You will be prompted:

Enter File Name:

Enter the name of the file you wish to unlock and press RETURN. The file will be unlocked, and the asterisk will be removed from that file name in the catalog. The file may then be deleted, replaced or renamed as usual. You may enter another selection from the DOS Commands Menu or press RETURN to get back to the text in memory.

Selection 5. Delete File: To delete a file stored on diskette enter 5 after the "Enter Selection" prompt. Your screen will display the message:

Enter File Name :

Enter the name of the file you wish to delete and press RETURN. The file stored under the name you entered will be deleted. The space occupied on the diskette by that file will be made available for saving other files.

CAUTION! There is no way you can recover a file which has been deleted. Check the file before using the delete selection (for example, by using the reverse slash [L]oad option to display the file on your screen) to make sure you are deleting the correct file.

Selection 6. Initialize diskette: Apple Writer][allows you to format a data diskette so that Apple Writer files may be saved on the diskette. This function has an advantage over the "Initialize Diskette" process explained in The DOS Manual in that 8192 additional characters of information can then be saved on your data diskette. However diskettes formatted as explained in The DOS Manual are completely compatible with Apple Writer][as well.

>>> TABULATION

Apple Writer][has nine tab stops already set at every eighth position. These are the default positions which are set when you "turn on" Apple Writer][.

CTRL-I: You may press [I] to automatically tabulate to the next available tab position on the line being entered. Pressing the [I] key enters blank spaces in the text up to the position of the next tab setting if the cursor is in the text entry mode (flashing blank). However, if the cursor is in move mode (flashing @), then [I] will skip over entered characters.

Data Line "Tab:" When entering text, you can determine the tab position of the cursor by looking at the number following "Tab:" on the Data Line. This number is the number of characters from the last RETURN to the position of the cursor. TAB POSITIONS ARE COUNTED FROM THE LAST ENTERED CARRIAGE RETURN, NOT THE LEFT SCREEN EDGE. This makes it possible to tab to locations on business forms that are wider than the screen display.

CTRL-T [T]ab(Set/Clear/Purge) : The [T] command permits the setting, clearing or purging of up to 32 tab positions. To set your own tab stops, press the space bar (or put the cursor in move mode and use the move keys if you want to avoid entering spaces while you are setting your tab stops) until the Data Line display shows the desired "Tab" number, then press [T]. Your screen will display the following prompt:

[T]ab(Set/Clear/Purge) :

OPERATION OF THE TEXT EDITOR

Enter an S (S = Set). The prompt line will disappear and the cursor will retain its position. The tab setting has been set for you and will be available using [I] as explained above.

To clear a single tab setting, move the cursor to the desired position (using the [I] key), then press [T], and in response to the prompt, press C (C = Clear). To clear ALL tab settings, press [T] and in response to the prompt, press P (P = Purge).

NOTE: Clearing memory ([N]ew) will NOT erase tab settings.

>>> SAVING AND RETRIEVING TAB SETTINGS

CTRL-Q2 Save Tab File: In some text editing applications, tab settings can be elaborate and difficult to duplicate. To save the tab settings you have developed, press [Q]. The "Additional Functions Menu" will be displayed on your screen. Enter selection number 2. The prompt:

Enter file name :

is shown at the bottom. Enter an appropriate file name, for example:

MYTABS,D2

Your tab settings will be saved under the file name "MYTABS" on the diskette in drive 2.

CTRL-Q1 Load Tab File: To retrieve a previously created tab file, press [Q]1. You will be prompted:

Enter file name :

Enter the file name of the desired tab list, for instance MYTABS,D2. The tab list will be loaded and the tab settings will become available, erasing any previous settings that may have been entered.

Tab File Name Prefix: The tab settings saved to diskette have file names with the prefix "TAB," for identification purposes. THIS PREFIX IS AUTOMATICALLY ADDED BY APPLE WRITER][AND YOU SHOULD NOT TYPE IT IN WHEN LOADING AND SAVING TAB FILES.

Changing Default Tab Settings: The default tab stops which are set when you first "turn on" Apple Writer][are stored on the master diskette under the file name "SYS". To change these default tab stops, first purge the present settings (press [T]p) and set the desired stops using the procedure explained above. With your Apple Writer][master diskette in drive one, use the Additional Functions Menu option 2 (press [Q]2) to save the new tab settings. In response to the "Enter file name:" prompt, enter:

SYS,D1 and press RETURN

The new tab stops will be set automatically each time you turn on Apple Writer][.

>>> "SEEING" RETURNS

There is one character in the text that is generally not visible. This character is called the RETURN. The RETURN doesn't print or display anything. Instead, it commands the cursor to move down to the next line and to the left edge, just like the "Carriage Return" on a typewriter.

When using Apple Writer][, it isn't necessary (or desirable) to end each line on the screen with a RETURN. Instead, end only PARAGRAPHS with RETURNS. The editor program will automatically format each text line for printing, so there is no need to worry about breaking the text into individual lines.

CTRL-Q6 Toggle Carriage Return Display: Sometimes it is necessary to see the RETURNS that have been entered in the text (when determining why the text you printed looks different than you expected, for example). To display RETURNS, you must do the following:

1. Disable the word wrap-around feature by pressing [Z],
2. Then press [Q]6. Now the position of each RETURN will be displayed as "]".
3. To return to the normal display, press [Q]6, then press [Z] again.

>>> UNDERLINING

NOTE: The underline feature as described in this manual will work on the QUME and Silentype printers. If you are using a different printer, check its manual for the command characters necessary to enable underlining.

Reverse Slash: The reverse slash (\) character has been set aside to act as an underline symbol. It is obtained by pressing ESC or [K] and then SHIFT-CTRL-P. A segment of text that begins and ends with this character will be underlined during printing. The underline symbol should occupy the place normally occupied by a space in the text, so that the text will be formatted correctly during printing. The entry:

The\reverse slash\also generates a space
would be printed:

The reverse slash also generates a space

NOTE: The printer you are using may require a different symbol to enable the underline feature. Check your printer manual to make sure you are using the correct symbol.

Note in the example above that the space between "reverse" and "slash" was not underlined. To force the printer to underline a blank space, an underscore character (SHIFT-CTRL-M) must be entered at the position the space would occupy. For example, the entry:

\Use_the_underscore_character_to_underline_a_blank_space.\
would be printed:

Use the underscore character to underline a blank space.

IMPORTANT!! AFTER THE REVERSE SLASH HAS APPEARED IN THE TEXT, UNDERLINING WILL CONTINUE UNTIL ANOTHER REVERSE SLASH APPEARS TO DISABLE THE UNDERLINE FEATURE.

NOTE: See the "Special Characters" section of the manual for a summary of the keystrokes necessary to obtain special characters.

>>> MORE ON TEXT DELETION AND RETRIEVAL

Apple Writer][offers a way to move words and paragraphs very quickly. This is one of the editor features that uses the direction arrow displayed on the Data Line. When the arrow is pointing to the left {<}, words or paragraphs can be DELETED and saved in a storage buffer. When the arrow is pointing to the right {>}, the words or paragraphs just deleted can be INSERTED, allowing you to easily move text from one part of the document to another.

CTRL-D [D]irection of Data Line Arrow: To change the direction of the arrow displayed on the Data Line, press [D].

CTRL-W [W]ord Deletion/Retrieval: Apple Writer][defines a "word" as a string of characters preceded by a space. To delete one of these words,

1. Position the cursor to the right of the LAST character of the word to be deleted.
2. Press [D] until the Data Line arrow is pointing to the left {<}.
3. Now press [W]. The word and the space preceding it are deleted and saved in a storage buffer.

To retrieve the word just deleted,

1. Move the cursor to the spot where you want to insert the word.
2. Press [D] to change the Data Line arrow to >.
3. Press [W]. The word and its preceding space will be inserted to the left of the cursor as though you had typed it in.

You can delete and move entire sentences using [W] multiple times. Just remember that the LAST word deleted will be the FIRST inserted. (The same technique is used when deleting and retrieving using the left and right arrow keys.)

CTRL-X Paragraph Deletion/Retrieval: The editor defines a "paragraph" to be a text string preceded by a RETURN. To delete a paragraph or a group of words preceded by a RETURN, move the cursor to the right of the LAST character in the paragraph or word group. Press [D] until the Data Line arrow is pointing left {<}. Press [X]. The entire text string (up to 1024 characters - about 12 lines of text on the screen) along with its preceding RETURN is deleted and saved in a storage buffer. The paragraph or word group just deleted may be inserted to the left (in "front") of the cursor by pressing [D] until the Data Line arrow points right {>}, and then pressing [X].

If you attempt to delete a paragraph which is longer than 1024 characters, the deletion will be accomplished up to the 1024th character, and a buzzer will sound indicating the storage area is full. In cases when the paragraph is longer than 1024 characters, move it in segments, or use the special SAVE and LOAD functions explained earlier.

NOTE: The length of a paragraph or any text string preceded by a RETURN can be determined by moving the cursor next to the last character and checking the number which is displayed opposite "Tab:" on the Data Line. This number gives the count of the characters entered since the last RETURN.

CTRL-R [R]eplace: [R] is an editor feature which allows you to change characters already entered without having to delete the unwanted text first. To enable this option, press [R]. An "R" will be displayed on the Data Line to remind you that you have enabled this special feature. Move the cursor in front of the character(s) you wish to change and enter the desired character(s). The old character will be replaced by the one just entered.

NOTE: Characters deleted in this fashion are not saved.

To disable [R]eplace, press [R] again. The "R" on the Data Line will disappear.

>>> AUTOMATED TEXT SEARCH AND REPLACEMENT

[F]ind is a powerful tool for finding and changing words and sentences in the text. [F]ind searches the text in memory FROM THE POSITION OF THE CURSOR AND IN THE DIRECTION OF THE DATA LINE ARROW.

[F]ind - Search Only: You may use the [F]ind feature to search quickly through your text for occurrences of a word or phrase. Move the cursor to the beginning of your file (press [B]), then press [F] and in response to the prompt "[F]ind :", enter the string you are searching for between two slashes. For example, to find the places in your text where the word "things" appears, enter:

 /things/ and press RETURN

In the example above, the two delimiters {/} mark the beginning and end of the search entry.

The cursor will position itself in front of the first occurrence of "things" that it finds, and the following message appears at the bottom of your screen:

 [F]ind:RETURN=Proceed

To stop the search, press any keyboard character other than RETURN. To continue to search the text for the next occurrence of "things", press RETURN. At each RETURN the editor will continue to search the text for the next occurrence of the word or phrase you have defined. If none are found, you will see the cursor displayed at the end of the file (if you searched forward from file beginning). The cursor will be displayed at the beginning of the text in memory if you searched backward from the end of the file.

CTRL-F [F]ind Text String and Replace: Suppose you have a large file in memory and want to find examples of the word "things" and replace some of them with the word "stars". Here's how:

1. Press [B] to move the cursor to the beginning of the text file. Remember, [B] also sets the Data Line arrow to >.
2. Press [F] . In response to the prompt, enter:

```
/things/stars/
```

This entry instructs the editor to find examples of the first word (text enclosed by diagonals) and gives replacement text to be used at your command.

3. Press RETURN. The editor will search for the word "things" and will stop at the first occurrence it finds. If it doesn't find any, you will see the cursor displayed at the end of the text.
4. If the editor DOES find the word, it will stop with the cursor next to the word and the following message is displayed:

```
[F]ind: RETURN=Proceed / Y=Replace
```

Now you may replace the first word {things} with the second {stars} by pressing Y, or not make the replacement and continue the search by pressing RETURN, or abandon the search and the [F]ind feature by pressing any other character.

If you elect to make the replacement, press Y. The editor will prompt you to press RETURN to continue the search. This sequence can be repeated until all occurrences of "things" have been found. Again, you may abandon the [F]ind feature during the search by pressing any character except RETURN or Y.

[F]ind and Replace Automatically: There is another way to use [F]ind. In this method, you want to replace one word (or letter, or sentence) with another, but you don't want to examine each occurrence that is found. To accomplish this, place an "A" after the third diagonal {/}. Here is a sample entry:

```
[F]ind :/things/stars/A
```

The "A" means "Replace ALL occurrences Automatically". The editor will replace all cases of "things" with "stars" without waiting for your decision on each case.

[F]ind - Replace Something with Nothing: It is also possible to delete text using [F]ind by not entering anything between the second and third delimiters. For example:

```
[F]ind :/naughty word//A
```

will delete ALL occurrences of "naughty word" since there are no characters between the second and third diagonals.

Entries to the [F]ind feature look like this:

```
[F]ind :/text to find/replacement/A
```

If the "A" (A=All) at the right of the entry is not included, the editor will stop at each occurrence of "text to find" and await a decision from you. If you press Y, the replacement will be carried out. If you press RETURN, the displayed word will be left as it is and the search will continue. If you type anything else, the search and the [F]ind feature will be abandoned.

If the Data Line arrow is pointing left {<} the [F]ind feature will search backward from the present cursor position. If the arrow is pointing right {>} the editor will search forward from the present cursor position.

The limit of the length of the entries using [F]ind is about 1-1/2 lines on the screen display. Here is an acceptable example:

```
[F]ind :/is very good friends with/has never met and  
doesn't know/A
```

[F]ind := Repeat the Previous Search: To repeat the last instruction used with the [F]ind option, press [F], then =.

[F]ind - An Example Application: [F]ind is a very powerful feature. It makes possible the use of "typing shorthand" that later may be converted into "longhand". Here is an example:

1. You have decided to type "*" instead of a difficult-to-type word or sentence. You create the document using this character everywhere the word or phrase is needed.
2. Then you move the cursor to file [B]eginning or [E]nd and enter:

```
[F]ind :/*Phineas Q. Phlagel/A
```

After you press RETURN, [F]ind will replace all "*" with "Phineas Q. Phlagel" automatically.

There is a way to automate even this process so that an entire shorthand language can be translated automatically using a computer language called WPL. This procedure is explained in the section of the manual on Word Processing Language.

NOTE: See the "Creating Your Own Glossary" section of the manual for another shorthand method.

KEYSTROKE SUMMARY OF [F]ind COMMANDS

- [F]/text to find/ Searches text in memory for first occurrence of "text to find". To continue search, press RETURN. Search may be exited by pressing any character except RETURN.
- [F]/text to find/replacement/ Searches text in memory for first occurrence of "text to find". If Y is pressed, "text to find" is replaced with "replacement". To continue search, press RETURN. Search may be exited by pressing any character except Y or RETURN.
- [F]/text to find/replacement/A Replaces ALL occurrences of "text to find" with "replacement".
- [F]/text to delete// Searches text in memory for first occurrence of "text to delete". If Y is pressed, "text to delete" is deleted. To continue search press RETURN. Search may be exited by pressing any character except Y or RETURN.
- [F]= Repeats the previous [F]ind instruction.

NOTE: [F]ind searches FROM THE POSITION OF THE CURSOR AND IN THE DIRECTION OF THE DATA LINE ARROW. To make sure you are searching your entire file, position the cursor at the beginning or end using [B] or [E], and check the direction of the Data Line arrow.

>>> USING [F]IND WITH SPECIAL DELIMITERS

In the previous examples of [F]ind, the delimiter "/" was used in the examples to separate the various segments. Other punctuation marks may be used as delimiters with special results.

If the standard "/" delimiter is used, any characters may appear between them, except the delimiter itself, and the editor program will correctly interpret the entry.

Using Other Delimiters: If another delimiter is used (a list is given after the examples), special features are enabled that use some characters as commands. In the examples which follow, special characters provide the following functions:

1. Delimiter {<}. This separates parts of the entry.
2. RETURN Symbol {>}. This is needed because the real RETURN key terminates the entry and begins the search.
3. Wild Card Symbol {?}. This symbol stands for any character so that only selected character positions are examined and others are accepted without examination.
4. Any Length Symbol {=}. Accepts text strings of up to 256 characters that have fields that need to be examined.

Here is a sample [F]ind entry using the special characters listed above:

```
[F]ind :<>Th??ty=><<
```

Translated into English, the preceding example means, "Find a phrase or word in which a RETURN {>} is followed by Th, then any two characters {??}, then ty, then any length {=}, and then another RETURN {>}. Replace this text string with nothing {<<} (ie. delete it) if I specify by pressing Y."

The wild card symbol {?} used in the search portion of the instructions (between the first and second delimiter) means "accept any character". In the replace portion (between the second and third delimiter) the wild card means "use the existing character, don't change it."

Here is another example using the same delimiter and special characters:

[F]ind :<??/??/79<?????80<

This means "Find the date format {??/??/79} and replace only the year {?????80}. Leave everything else as it is (using the ? as a wild card).

Possible Delimiter Selections and Associated Special Symbols:

Suppose you want the described features, but you need to use "?", "<", or "=" as normal characters in your entry. To do this, you must select a different delimiter. Here are the acceptable sets:

<u>Delimiter</u>	<u>Any Length Symbol</u>	<u>RETURN Symbol</u>	<u>WILD CARD Symbol</u>
/	None	None	None
<	=	>	?
#	\$	%	&
&	'	()
!	"	#	\$
*	+	,	-

>>> CREATING YOUR OWN GLOSSARY

Often certain words or phrases are difficult to type quickly. Using the [G]lossary feature of the editor, you can eliminate this problem. [G]lossary offers you the capability of creating a long list of such words and phrases and will insert these text strings into the text file with a single keystroke.

Defining Glossary Terms: Using [G]? you may define text strings by entering a single character followed by the desired text. The text string may then be inserted in the file in memory at any time by pressing [G] followed by the character used to define the text string. For example, suppose you were creating a document which repeatedly used the phrase "Phineas Q. Phlagel, Inc.". To define this phrase, press [G]. The following prompt will appear on your screen:

[G]lossary(=?=Define/*=Purge) :

Enter a question mark. (DON'T PRESS RETURN.) Your screen will immediately display the prompt:

Enter new definition :

Precede your phrase with a single character. For example, your entry might look like this:

Enter new definition :pPhineas Q. Phlagel, Inc.

and press RETURN to terminate the entry. The first character entered in response to this prompt ("p" in this example) defines the text which follows, and will be the character you use to recall the text for insertion. Each time you use [G]? to enter a new glossary definition, you will see any previous definitions at the top of your screen, followed by the prompt, "Enter new definition :".

The number of glossary terms you define is limited only by the number of characters on the keyboard. Any character except "?" and "*" may be used. Upper case and lower case letters are regarded as different characters by the [G] function. For example, you may use an upper case "A" to define one glossary term and a lower case "a" to define a different term.

The TOTAL LENGTH of all the glossary definitions may not exceed 2048 characters. A SINGLE TERM may not exceed 128 characters when defined using [G]?. See "Saving a List of Glossary Definitions" below for an alternate procedure for defining glossary terms.

Inserting [G]glossary Terms into Text: Once you have defined a glossary term, you may insert it into a document by pressing [G] followed by the character you used to define the string. The text will be inserted at the position of the cursor. Following the example above, whenever you wished to insert the phrase, "Phineas Q. Phlagel, Inc." in your text, you would press [G]p. However these definitions are lost if the computer is turned off.

Saving a List of Glossary Definitions: If you are creating a document or series of documents which repeatedly use the same group of phrases, you will want to save yourself the trouble of entering your glossary list every time you "turn on" Apple Writer][by creating a glossary file. The procedure to follow is:

1. Clear memory ([N]). Type in each definition just as you would normally enter text, entering the single character first, and then your phrase. Terminate each definition with a RETURN. For example,

```
pPhineas Q. Phlagel, Inc.
3third quarter budget requirements
Pprogress reports for the period
```

The length of the file you create cannot exceed 2048 characters. However, the only limitation on the length of each definition is the total file length. In other words, you could create a single glossary term of 2048 characters. To check the length of your file, look at the number displayed next to "Len:" on the Data Line. If the number is larger than 2048, you will have to delete some text in order to [L]oad the file using [Q]5 (explained below).

OPERATION OF THE TEXT EDITOR

2. [S]ave the list in the usual way. For example,

[S]ave :GLOSSARY1,D2

3. To load your glossary file, use the "Additional Functions Menu" (CTRL-Q), selection 5. You will be prompted:

Enter file name :

Enter the file name you used to [S]ave the glossary file, and press RETURN. The file will be loaded and available for use under the [G]-character function.

Deleting Glossary Definitions from Memory: To delete all glossary definitions in memory, press [G]*. This action will not delete glossary files [S]aved on diskette.

NOTE: Clearing memory ([N]ew) will NOT erase glossary definitions in memory.

KEYSTROKE SUMMARY OF [G]LOSSARY COMMANDS

- | | |
|-----------------------|--|
| [G]?{character}{text} | Defines a glossary term which can be inserted in text using [G]{character} |
| [G]{character} | Inserts a glossary term defined using [G]{character}{text} at the position of the cursor |
| [G]* | Deletes all glossary terms in memory. |

NOTE

A single glossary term defined using [G]? may not exceed 128 characters.

The total length of all glossary definitions may not exceed 2048 characters.

>>> SUMMARY OF THE DATA LINE DISPLAY

The first character on the Data Line is called the direction arrow and looks like <. Its function is to establish the direction of search from the cursor when using editor features such as [F]ind. It is also used to indicate whether you are inserting {>} or deleting {<} while using [W] or [X], the word and paragraph insert/delete commands. To change the direction of the arrow, press [D].

The second character indicates which of the three options for upper case entries is in effect: E means that the ESC key must be used to obtain upper case; S means that the SHIFT key can be used (provided the shift key modification has been made); and U indicates that all entries will be in upper case.

The third position on the Data Line will contain a "V" if the [V] Accept Control Characters feature is enabled, or an "R" if the [R]eplace feature is enabled. These features are explained in the "Control-V Control-characters as Text Entries", and "Control-R [R]eplace" sections of the manual.

The fourth character on the Data Line is an asterisk that appears if keyboard entries are being made faster than they can be shown on the screen. This doesn't mean that such entries will be lost; they are simply saved until they can be processed. The user may be typing as much as 64 characters ahead of the display (a very unlikely situation) and all the characters will be processed.

The fifth character which appears on the Data Line is a "Z". The "Z" is displayed when the word wrap-around feature is enabled. To disable word wrap-around, press [Z]. The "Z" will no longer be displayed on the Data Line. To enable word wrap-around, press [Z] again.

The next items you see are four abbreviations followed by numbers. Their meaning is as follows:

Mem	Free text memory remaining (the remaining number of characters which may be entered in the file in memory)
Len	Number of characters entered in memory
Pos	The character number of the cursor's current position (If the cursor is at the end position, Len and Pos will be the same.)

OPERATION OF THE TEXT EDITOR

Tab The number of characters from the last RETURN to the cursor's present position

The last entry on the Data Line is "File:" and contains the file name last entered when using [S]ave or [L]oad commands (see "Saving Your Text" and "Loading Files into Memory").

CTRL Q8 - DATA LINE TOGGLE: The keystroke sequence to suppress or display the Data Line is [Q]8. Pressing [Q] will cause your screen to display the ADDITIONAL FUNCTIONS MENU. The menu selection number which turns the Data Line on and off is 8.

>>> CONVERTING APPLE WRITER 1.1 FILES

If you have text files which were created using Apple Writer 1.1, you may convert them to Apple Writer][by using selection 9 on the Additional Functions Menu. Here's how:

1. Clear memory ([N]ew).
2. Insert the diskette containing the Apple Writer 1.1 file you wish to convert into one of your disk drives.
3. Enter [Q]9. You will be prompted:

Enter file name :

Enter the name of the Apple Writer 1.1 file followed by the number of the drive containing the diskette. For example:

MEMO,D1 and press RETURN

The Apple Writer 1.1 file will be loaded into memory.

NOTE: Do not type in the "TEXT." prefix which is automatically assigned to Apple Writer 1.1 files. Also, if you have files which were created using the original 13 sector Apple Writer 1.0, you will first need to convert them to 16 sector format using the MUFFIN program described in The DOS Manual.

4. To convert text-embedded print commands, insert your Apple Writer][master diskette in drive one and enter:

[P] DO CONVERT,D1

5. [S]ave the file on an initialized data diskette.

>>> SPECIAL FEATURES AND TECHNIQUES

CTRL-C Automatic [C]ase Change: Suppose you want to change the case of an entire word or line. Apple Writer][offers a feature which allows you to automatically change text from lower case to UPPER case or vice versa. This editor function is enabled by pressing [C]. The cursor will display a U. If you press [C] again, the cursor will display an L. Each time you press [C] the cursor will switch between U and L.

If you want to change some text from lower case to upper case, for example, you would:

1. Move the cursor next to the text you wish to change.
2. Press [C] until the cursor displays a U.
3. Move the cursor across the text to be changed using the "J", "K", "S", or "D" key.

The words passed over by the cursor will be changed to UPPER case.

To change UPPER case to lower case, press [C] until the cursor displays an L, and follow the same procedure.

To disable [C], press the space bar to begin entering text, or the ESC key to continue cursor movement mode.

CTRL-Y Split Screen Operation: Sometimes it is necessary to see and make changes in widely separated parts of the text file at the same time; for example, while moving text segments around. The [Y] feature allows you to horizontally split the normal 24-line display into two different 12-line displays. Here's how:

1. Press [Y]. You will be prompted:
[Y] Split Screen: (Y/N) or (RETURN) Switch
2. To enable the split screen feature, press Y and RETURN.
3. To switch cursor control from one screen display to the other, press [Y], and in response to the prompt, press RETURN.
4. To disable the split screen, press [Y], and in response to the prompt, press N and RETURN.

The screens have independent cursor positions. You may set one cursor to a position in the file where you want to move text, and use the split screen feature along with text deletion and insertion to move text segments from place to place. You may also use split screen to view a previous text version (recalled from disk using [L]oad with the reverse slash suffix) while looking at its present form at the same time. You may also enter text while the split screen option is enabled.

NOTE: Any changes made to the text file while using [Y] will be accepted as usual. [Y] DOES NOT CREATE TWO IDENTICAL TEXT FILES NOR ALLOW EDITING OF TWO DIFFERENT FILES AT ONCE; YOU ARE SIMPLY LOOKING AT TWO SEPARATE LOCATIONS OF THE SAME FILE.

CTRL-V CTRL-characters as Text Entries: The editor normally interprets CTRL-characters as commands. You may also enter CTRL-characters as text file entries for purposes such as controlling special printer features. (See the sections below on Footnotes, and Super/subscripts.) To enter CTRL-characters as text:

Press [V]. A "V" will be displayed on the Data Line to indicate that the option has been selected. CTRL-characters will now be accepted into the text file. While [V] is enabled, the CTRL-characters will no longer have their command functions. The CTRL-characters will be displayed as inverse (black on white) upper case letters.

To disable [V] and return to normal CTRL operation, press [V] again.

Superscripts and Subscripts: In some documents it is necessary to refer to footnotes or a bibliography using characters that are above or below the normal printed line. To accomplish this, refer to your printer manual for the instruction sequences that permit this and enter them using the "[V] Accept Control Characters" feature.

On the Qume printer, for example, a superscript is preceded by a reverse half line feed (ESCAPE SHIFT-D) and followed by a normal half line feed (ESCAPE SHIFT-U). For a subscript, the order is simply reversed. Because these characters are not printed, although they occupy spaces in the formatted line, they should be used in left justified text rather than fill justified for best results. (See the "Printing the Text File" section of the manual for a full explanation of the printing options.)

NOTE: The platen release lever on your printer must be engaged when printing super/subscripts. This provides tension between the paper and the roller so that the paper can be moved backwards.

Footnote Entry Procedure: Many text editing tasks require the inclusion of footnotes. There is an entry procedure for footnotes that will automatically move and print footnote text at the bottom of the page when the document is printed. Any text which is preceded by an open parenthesis and a less-than character { (<) and ended with a greater-than character and a close parenthesis {>} will be printed at the bottom of the page. Here is an example which uses the Qume Sprint 5 commands:

As entered:

```
.. in other cases[V] ESCAPE SHIFT-D [V] 3 [V] ESCAPE
SHIFT-U [V] (<3. Wild & Untamed, Putnam 1968> ) reported ..
```

As printed:

```
.. in other cases3 reported ..
```

(with the footnote text printed at the bottom of the page)

NOTE: See "Appendix B Special Characters Glossary File" for a much easier entry procedure of subscripts and superscripts.

In the example above, the two characters (< identify the beginning of the footnote text. The two characters >) identify the end of the footnote and must be followed by an extra space.

CAUTION!! These two-character symbols are reserved for footnote identification and should not be used for other purposes.

The usual footnote entry procedure would be to enter the footnote text immediately after its identifying superscript. This practice will reduce the possibility that the footnote might be printed on a succeeding page.

Footnotes will be printed at the left margin. To indent the footnote, enter spaces after the "<" symbol.

NOTE: To ensure that footnotes which fall on the last page of a document are printed, a form feed command must be embedded at the end of the text file. See the instructions under "Form Feed" in the "Printing The Text File" section of the manual.

>>> SPECIAL CHARACTERS

There are some special characters which are not shown on your keyboard such as the backslash and underscore. To enter these characters follow the keystroke sequence given in the table below.

SUMMARY OF SPECIAL CHARACTERS

KEYSTROKE SEQUENCE	KEYBOARD OPTION (2nd Character on Data Line)				
	E	S	S*	U(E)	U(S)
SHIFT M	}	}	M]]
SHIFT N	~	~	N	^	^
SHIFT P	`	nothing	P	@	@
CTRL SHIFT M	_	_	_	_	_
CTRL SHIFT N	{	{	[[[
CTRL SHIFT P			\	\	\
ESC SHIFT M]	down 1 line	down 1 line]	nothing
ESC SHIFT N	^	nothing	nothing	^	nothing
ESC SHIFT P	@	nothing	nothing	@	nothing
ESC SHIFT [P]	\	nothing	nothing	\	nothing

*Shift-key modification installed

See "Appendix B Special Characters Glossary File" for a simple procedure for entering these special characters.

NOTE: These characters will be printed on most printers as shown above. However some of the characters will be displayed differently on your screen depending on whether you have the 80-column board.

>>> PROTECTING YOURSELF AGAINST DISASTER

The computer's memory contents are lost if power is removed from the computer. Therefore, it is a good idea to [S]ave your work frequently. Every 10 minutes is NOT too often.

For important work, make sure that the file was saved correctly. Here's how:

1. [S]ave the file to disk

```
[S]ave :FILENAME,D2
```

2. [L]oad the same file name to your screen using the reverse slash suffix

```
[L]oad :FILENAME,D2\           (reverse slash option)
```

so that you may examine it (without disturbing the present memory contents) to determine if the [S]ave operation was successful.

3. Also consider saving the file to another diskette as well.

The diskettes themselves are not perfect. Things can go wrong with them. They might be exposed to excessive heat or static electricity, cigarette ashes or spilled coffee. Therefore, it is a very good idea to make plenty of backup copies of important files. See The DOS Manual for the procedures to follow to make copies of your diskettes.

>>> DISK ERROR MESSAGES

While using the [S]ave and [L]oad features, it is possible to see an error message displayed at the bottom of the screen. An error message looks like this:

```
DOS: File not found
```

After reading the message, press RETURN and Apple Writer][will resume at the Editor Menu. Some error messages are more serious than others. For example, the message:

DOS: : I/O ERROR

means that there is no diskette in the disk drive, or, if there IS one present, it may be destroyed. See The DOS Manual for more information about error messages.

>>> IN CASE OF TROUBLE

Computers sometimes misbehave. This might be brought on by a room temperature that is too high or low, a static electricity buildup caused by low humidity or synthetic rugs and fabrics in the office. The symptoms of reversible computer errors are a normal display but no response to the keyboard, or erratic operation. Here are some rules to follow to maximize your chances of recovering memory contents:

FIRST: DON'T TURN THE COMPUTER OFF. Turning the computer power off will destroy any memory contents not saved to disk.

SECOND: If the operation preceding the failure involved [S]ave, [L]oad, or [P]rint (to be explained in the next section), try pressing the RETURN key to exit the function.

THIRD: Press RESET (or CTRL-RESET on newer model Apple][Computers). Apple Writer][is designed to preserve your file when RESET is pressed. If the RESET sequence has been successful, the Editor Menu will appear, as when the system is turned on. In most cases, the file will be preserved in memory and may be saved.

FOURTH: Protect yourself!! See the very important section on "Protecting Yourself Against Disaster".

>>> QUITTING APPLE WRITER]]

When you are through using the Apple Writer]] program, you may quit Apple Writer by selecting option 0 (zero) on the "Additional Functions Menu".

CTRL-Q0 Quit Apple Writer: If you press [Q]0, you will be prompted:

Erases Memory. Quit (Y/N) ?

You now have the opportunity to change your mind. This is the time to ask yourself if you have [S]aved the file in memory and are really ready to lose it. If the answer is "yes", enter a Y and press RETURN. ALL ENTRIES IN MEMORY ARE NOW LOST.

You may now insert the system diskette which you want to use next in drive one and enter PR#6 to reboot. If you have the 80-column board, the screen will be blank and you will need to disconnect the cable from the monitor and connect the alternate cable for 40-column operation.

This section of the manual explains the printing of text files created using the Apple Writer][Text Editor. It is important to realize that some experimentation will be necessary to acquire printing skill. Ten sheets of paper used in practice at the beginning will save hundreds later on.

There are many printing format choices available. Most of them may be made before printing is underway. Some of the choices may even be changed during printing by means of printing commands "embedded" in the text file being printed.

Selections such as left and right margin settings, form length, page number, page heading and so forth, may be made before printing commences. These values are called "format values".

PRINTING THE TEXT FILE

>>> PAGE FORMAT COMMANDS

CTRL-P ? [P]rint/Program Commands-Display: To see a list of the present format values, press CTRL-P. At the bottom of your screen you will see the message :

[P]rint/Program :

Now press ? and RETURN

The entry "?" directs the editor program to display the list of "Print/Program Commands". A list of formatting options with letter codes and numbers will appear. Next to each format option is a two-letter code in parentheses followed by a number or characters as shown below.

```
Print/Program Commands:
Left Margin (LM) = 9
Para. Margin (PM) = 9
Right Margin (RM) = 75
Top Margin (TM) = 1
Bottom Margin (BM) = 1
Page Number (PN) = 1
Printed Lines (PL) = 58
Page Interval (PI) = 66
Line Interval (LI) = 0
Single Page (SP) = 0
Print Dest. (PD) = 1
Car. Return (CR) = 0
Print Mode
(L) (L) (P) (R) = LJ
Top Line (TL) :
Bottom Line (BL) :
[P]rint/Program :
```

Until you enter your own values (procedure explained below), the "Default Value", provided by the program, will be used. To leave the display and return to your text file at the original cursor position, press RETURN.

Format Command Codes and Default Values: Here is an explanation of each code and its default value:

Left Margin (LM) = 9 This number selects the left margin setting. The printing of each line begins at the specified number of spaces from the left edge.

Paragraph Margin (PM) = 0 Indents or outdents the first line following a RETURN by the number of spaces specified. The value of "PM" is relative to the left margin setting.

Right Margin (RM) = 79 The printing of each line ENDS at the specified number of spaces from the left edge.

Top Margin (TM) = 1 This is the number of blank lines between the page top line entry ("Top Line" discussed below) and the first printed line.

Bottom Margin (BM) = 1 This is the number of blank lines between the bottom of the last printed line and the page bottom line entry ("Bottom Line" discussed below).

Page Number (PN) = 1 This number selects the beginning page number to be printed, if desired, on the page top line or bottom line entry. The page number is increased by 1 automatically for each successive page printed. To print the page number see "Top and Bottom Line Entry Procedure" below.

Printed Lines (PL) = 58 This number selects the number of text lines per page. Printed lines includes Top Line and Bottom Line entries, the blank lines specified by "TM" and "BM", lines actually carrying text, footnotes, and blank lines created by RETURNS.

Page Interval (PI) = 66 This number selects the FORM LENGTH or how many blank lines would fit on a page. When using a standard printer and 8-1/2 by 11 inch forms, the default value of 66 is correct.

PRINTING THE TEXT FILE

Line Interval (LI) = 0 This number selects the number of blank lines between each printed line. A value of 0 is used for single-spacing, 1 selects double-spacing and so forth.

Single Page (SP) = 0 This entry selects/deselects single page printing. If the Single Page option is selected (1="YES"), the program will prompt you to insert a new sheet of paper into the printer at the completion of each printed page. If the Single Page option is deselected (0="NO"), it is assumed that continuous "fanfold" or "roll" paper is in use and there is no need to manually insert paper into the printer.

Print Dest. (PD) = 1 This entry selects the slot in which your printer interface card is inserted. The default entry "1" will cause your file to be printed on a printer connected to an interface card in slot 1. An entry of "0" will "print" your text on the video screen.

NOTE: When "printing" to the screen, set "LM" to 0 and "RM" not greater than 79 (39 if you don't have an 80-column board).

Carr. Return (CR) = 0 This entry determines whether a "line feed" signal will be sent to your printer. If your printer does not automatically move the paper sheet up one line each time it begins to print a new line of text, you will need to change this value to "1". (Procedure explained below.)

You can tell if your printer is moving the paper up one line by printing some text with "CR" set to zero. If text is printed on separate lines, leave "CR" set to zero. If all text is printed on the same line, set "CR" to 1.

Print Mode (LJ,FJ,CJ,RJ) = LJ This format option is a specifier for "justifying" your text. LJ means "left justify", FJ means "fill justify", CJ means "center justify", and RJ means "right justify". (See "Justify Options" below.)

Top Line (TL): This is a text entry. The entered line will be printed at the top of each page and may appear at the left, center, or right of the printed page, or all three (see "Top And Bottom Line Entry Procedure" below).

Bottom Line (BL): This is a text entry like "Top Line" which will be printed at the bottom of each page.

>>> CHANGING FORMAT VALUES USING CTRL-P

There are three methods you may use to enter new format values. The first allows you to see the "Print/Program Command" list as you enter the new values. You will also be able to change values while in the text entry mode. And, finally, you will be able to "embed" commands within your text to change format values while printing is actually underway. This last technique is discussed in the "Text Embedding Printing Commands" Section of the manual.

[P]? - Using the Print/Program Commands Display as a Menu:
In addition to serving as a reference guide for format commands, the "Print/Program Commands" display also serves as a menu for changing format options. To use the menu:

1. Press [P], and in response to the "[P]rint/Program: " prompt enter a question mark and press RETURN. The "Print/Program Commands" list will be displayed on your screen. At the bottom of the list you will again see the "[P]rint/Program: " prompt.
2. Enter the two-letter code (called "command characters") of the format value you wish to change immediately followed by the desired value. (When changing Print Mode, simply enter LJ, FJ, CJ, or RJ). For example, to change the left margin setting to 15 you would make this entry:

[P]rint/Program :LM15 and press RETURN

The "Print/Program Command" list will be updated and display the value you just entered. The prompt remains at the bottom of your screen, and you may now enter another new value by entering the appropriate command characters followed by the desired value and pressing RETURN. Follow this procedure until you are finished making entries. Press RETURN again, and the editor will return you to the text file in memory at the original position of the cursor.

PRINTING THE TEXT FILE

NOTE: It is not necessary to enter the command characters in UPPER CASE.

In the example above, the value "15" is called an "absolute" value. You may also make changes to the format values by entering a plus or minus followed by a number. This is called a "relative" value. For example, to change the left margin from its default value of 9 to 15, you could enter:

```
[P]rint/Program :lm+6
```

instead of the example entry shown above. Relative value settings are most useful as text-embedded print commands.

[P] Changing Format Values Without Using the Menu: If you remember the two-letter command code of the format value you wish to change and want to save yourself some time, you may change format values without using the menu by simply pressing [P], then entering the two-letter command code followed by the desired value. For example, if you wanted to change the right margin setting to 90 without using the menu, you would press [P] and enter:

```
[P]rint/Program :rm90          and press RETURN
```

>>> TOP AND BOTTOM LINE ENTRY PROCEDURE

Top line and bottom line entries contain information such as page number, titles, and dates. The text that you wish these lines to contain is entered using [P]rint/Program Commands "TL" and "BL" (the last items on the Print/Program Commands list). To enter the desired text, press either [P]? (to see the Print/Program Commands list) or [P], followed by either TL (for top line entries) or BL (for bottom line entries). THE NEXT CHARACTER ENTERED will be considered as a delimiter to separate the parts of the entry which will be printed on the left, center, and right of the top or bottom line.

PRINTING THE TEXT FILE

Suppose, for example, that you wanted "1st Quarter Sales Report" to be printed at the top left of every page, the page number at top center, and the date at top right. Your entry would look like this:

```
[P]rint/Program :TL*1st Quarter Sales Report*Page #*4/15/1981*
```

In this example the "*" serves to separate (delimit) the text segments which appear *left*center*right*.

The "#" sign in an entry directs the page number to be printed at its location. (See "Printing Page Numbers" below.)

CAUTION!! There are two characters which may not be used as delimiters. They are the reverse slash, which commands the printer to underline, and the # sign, which directs the page number to be printed. A space entered following TL or BL would be considered a delimiter by the editor.

If you decided that you would rather have "1st Quarter Sales Report" appear on the top right, and the date and page number on the bottom line your entries would look like this:

```
[P]rint/Program :TL***1st Quarter Sales Report*  
[P]rint/Program :BL*4/15/81*Page #**
```

The entry for Top Line uses the "*" character as a delimiter with no text between the first and second delimiter or the second and third delimiter since you want "1st Quarter Sales Report" to appear on the right. The entry for Bottom Line would cause "4/15/81" to appear on the bottom left and the current page number to appear on the bottom center of every page.

NOTE: To print a bottom line entry on the last page you must enter a "form feed" command at the end of the text file. This procedure is explained in the "Text-Embedding Printing Commands" section of the manual.

Printing Page Numbers: To print a page number, you must enter a "#" sign in a "TL" or "BL" entry. You can only direct page numbers to be printed in the "TL" and "BL" entries. The number of the page being printed will be positioned in the heading or footing according to the entry you make after TL or BL. The page number will NOT be printed if the "#" character is not entered in TL or BL. For example, the entry:

```
t1//##//
```

would cause only the number of the page being printed to be printed at the center of the top line. You may also have entries such as "t1//Page #//" or "t1//Page # of 20//", etc.

>>> OPTIONS FOR JUSTIFYING YOUR TEXT

You may choose from four different justify modes for printing your text. The default value, Left Justify, "LJ", will print your text as it normally appears when you are typing, flush at the left margin. Fill Justify, "FJ", will add spaces between words so that text lines are flush at both the left and right margin. This is how most magazines and books are printed. Center Justify, "CJ" will center any text on the line. This mode would be used for centering table headings and the like. Right Justify, "RJ", prints text flush at the right margin.

To change the justify mode, press [P]? or [P] followed by your choice of LJ, CJ, FJ, or RJ. You may also change the justify mode with a text-embedded command (see "Text-Embedding Printing Commands below). Here are some sample entries and results:

Left Justify [P]rint/Program :LJ

```
Here
are some
test lines to
show the effect of
the various printing modes.
```

PRINTING THE TEXT FILE

Center Justify [P]rint/Program :CJ

Here
are some
test lines to
show the effect of
the various printing modes.

Right Justify [P]rint/Program :RJ

Here
are some
test lines to
show the effect of
the various printing modes.

Fill Justify [P]rint/Program :FJ

Here are some test lines to
show the effect of the various
printing modes. The left
margin is set to 20 and the
right margin is set to 50. No
carriage returns are in this
text and all of the extra
spaces you see are supplied by
Apple Writer during printing
with Print Mode set to "FJ".

>>> TEXT-EMBEDDING PRINTING COMMANDS

Embedding Format Values: Any of the above printing commands may be entered (embedded) into the text file you wish to print. To embed a [P]rint/Program command, first position the cursor at the point in the text where you wish to make a format change. Press RETURN. Then enter a period immediately followed by the two-letter command code, then the new value and then another RETURN. Each embedded command must be on a separate line, preceded and followed by a RETURN. The capability of entering relative values is especially useful when text-embedding printing format commands. If you want to indent some text for emphasis, all you need to do is press RETURN and enter:

```
.lm+5      and press RETURN
.rm-5      and press RETURN
```

At the end of the text you wanted indented, follow the same procedure to return the format to its original values:

```
.lm-5      and press RETURN
.rm+5      and press RETURN
```

You may also enter absolute values. For example if the left margin is set to 9 and the right margin to 79, the paragraph could be indented by the following instruction:

```
.lm14      and press RETURN
.rm74      and press RETURN
```

Printing commands entered in this fashion are treated as instructions to the printer and will NOT be printed.

Indenting and Outdenting Paragraph Margins - .PM To indent the first line of a paragraph you may use .PM followed by the number which represents the number of spaces you wish to indent the FIRST LINE of each new paragraph. ".PM" is always "relative" to the left margin setting. ".PM+2" means the first line following a RETURN will be indented 2 spaces to the right of the left margin setting. If the plus sign is not included the number will be interpreted as a positive relative number. In other words, ".PM2" would have exactly the same effect as ".PM+2".

You may also use negative numbers. The entry `.PM-4` would "outdent" the first line following a RETURN four spaces to the LEFT of the left margin setting. For example, the embedded command `.PM-4` would cause text to print in the following format:

preceding paragraph text

1. new paragraph text
new paragraph text

The "1." was outdented 4 spaces, and the remainder of the paragraph was printed at the left margin value of "LM". To return to the previous format, enter `.pm0` on a separate line after the indented text.

If you wish the line beginning with "1." to be flush with the preceding left margin, the series of entries:

```
.lm+4
.pm-4
```

would cause your text to be formatted in this manner:

preceding paragraph text

1. new paragraph text
new paragraph text

To return to the previous format, enter `.pm0` and `.lm-4` on separate lines after the indented text.

Centering a Line: Another example of an embedded command would be to center a single line of text such as a title. The series of entries:

```
.cj
TEXT OF TITLE
.lj
```

would cause TEXT OF TITLE to be centered between the left and right margin settings and any text following it would be left justified.

>>> ACTUALLY PRINTING

[P] NP - New Print: There are two commands which actually get printing underway. The first is "New Print", NP.

NOTE: THE FILE TO BE PRINTED MUST BE IN MEMORY.

The text will be printed according to the format options which you have selected, and each page will be numbered successively beginning with the value of "PN". (The page number will not be printed unless specified in a "TL" or "BL" entry. See "Top and Bottom Line Entry Procedures".) Example entry:

[P]rint/Program :NP

Stopping the Printer: To stop the printer during printing, press ESCape. This action will also cause the Editor Menu to appear on your screen. Press RETURN, and you will be back at the beginning of the file in memory.

[P] CP - Continue Print: The other printing option is "Continue Print" or "CP". This option is used when you want to print more than one text file as a single document. Use the following procedure:

1. Set [P]rint/Program format options
2. Erase Memory: [N]ew: Y
3. [L]oad the first file: [L]oad :FIRSTFILE,D2
4. Start printing: [P]rint/Program :NP
5. When printing stops, erase memory: [N]ew: Y
6. [L]oad the next file: [L]oad :NEXTFILE,D2
7. Continue printing: [P]rint/Program :CP
8. Repeat steps 5 through 7 as required.

The [P] CP option will continue printing at the point on the page where the previous file ended and will continue to number pages successively.

NOTE: The described method for multiple-file printing is made simpler by using WPL, described in the Word Processing Language section of the manual.

Printing Only Selected Pages of a Document - .EP{0 or 1}
 This is a command which can only be embedded in the text file. If you only wanted to print pages 5 and 6 of a document (the only ones that had editing changes) you would enter at the beginning of the text file:

.EPO and press RETURN

The "0" after ".EP" will cause the printer to ignore (not print) any text which follows. (The editor program will keep track of the page number.) On the line before page 5 begins, enter:

.EPI and press RETURN

The printer will print any text which follows according to the format instructions given with the correct page number and headings. After the last line of page 6 you would again enter .EPO and the printer will ignore any following text.

When You DON'T Want a Text Line Printed: A text line will NOT be printed if:

1. It is preceded by a RETURN, and
2. the first character after the RETURN is a period.

Several such lines may be entered and the RETURNS that separate them won't be printed. Be careful that you don't unintentionally enter a format command!

Suppressing First Page "TL" and "BL" Entries: If you would prefer that your top line entries were not printed on the first page, you can embed the entries at the beginning of your text file instead of entering them under [P]. For example, if you wanted the heading:

1st Quarter Sales Report Page # April 15, 1981

to appear on every page except the first one, you would embed the following command at the beginning of your text file:

.tl*1st Quarter Sales Report*Page #*April 15, 1981*

and press RETURN.

The example heading would be printed on every page except the first.

To suppress the bottom line entry, embed the desired "BL" entry anywhere within the text which would appear on the second page.

NOTE: Text-embedded print commands alter the default values shown on the [P]rint/Program Commands menu. This allows printing of other files using the same margins by doing a [P] CP (Continue Print) command. To restore the default print values, enter [Q]3 and load the file named SYS.

>>> ADDITIONAL PRINTING COMMANDS

There are some additional printing commands which may only be text embedded.

Form Feed - .FF This command instructs printing to begin on a new page. Any top and bottom line entries which have been specified will be printed as usual. This command would be used at the end of chapters or whenever a new page beginning is needed.

NOTE: A ".FF" command must be used at the end of a document to instruct the printer to eject the page and to print footnotes and bottom line entries on the last page. However it should NOT be followed by a RETURN. A RETURN after ".FF" will cause the top line entry to be printed on the following page.

Form Feed Only If - .FF{number} This command may be used to conditionally feed a new sheet to the printer. For example,

.FF20

would cause the printer to skip to the top of the next page only if there are NOT at least 20 lines left on the page. This command may be used to make sure that there will be room for a table or some other text that would not look or read well if printed on different pages. In the example above, it is anticipated that 20 lines will be needed to print a table. The instruction .FF20 will insure that the entire table is printed on one page.

PRINTING THE TEXT FILE

Automatic Prompting - .IN This embedded command will stop printing and display a message on the screen. The entry (on a separate line):

```
.IN It's time to insert different paper.
```

will halt the printer, and display the text "It's time to insert different paper." on the screen. When the special task has been completed, press RETURN to continue printing.

Using the [V] "Accept Control Characters as Text" command, you may also command a buzzer to sound along with your automatic prompt. The text-embedded CTRL-character command [G] will cause the buzzer to sound if entered after ".IN". For example, the entry:

```
.IN [V] [G] [V] It's time to insert different paper.
```

would sound a buzzer in addition to displaying the prompt, "It's time to insert different paper."

>>> SAVING AND LOADING [P]RINT/PROGRAM COMMAND VALUES

You may save, then recall for use, a set of format values. All the values in the [P]rint/Program Command list may be saved for future use by using the Additional Functions Menu. Here is how:

1. Set the format values on the [P]rint/Program Command list to the desired values.
2. Press [Q]4. You will be prompted:

Enter File Name:

3. Enter a file name, for example:

MYFORMAT,D2 and press RETURN

The format values will be saved for you and you will be returned to the file in memory at the original position of the cursor.

To recall the set of format values you saved:

1. Press [Q]3. You will be prompted:

Enter file name :

2. Enter the file name you used to save your format values and press RETURN. The values will replace any that had previously been set.

NOTE: [P]rint/Program Commands values saved using this feature have the prefix "PRT." automatically added to the file name for identification purposes. This prefix should not be entered by the user when loading and saving format files.

Permanently Changing Default [P]rint/Program Values: You may wish to "permanently" change the [P]rint/Program command default values if you find you are making the same format value entries each time you turn on Apple Writer][. The [P]rint/Program Commands default values are stored on the Apple Writer][master diskette under the file name SYS . To change these values to ones which will better suit your needs, make the desired entries as usual. Make sure your Apple Writer][master diskette in drive one. Use the Additional Functions Menu selection 4 (press [Q]4), and in response to the "Enter file name : " prompt enter:

SYS,D1

The [P]rint/Program Commands values saved in this manner will be the new default values each time Apple Writer][is used.

NOTE: The Shift Key Option setting ("E" or "S" on Data Line) is also saved with the [P]rint/Program values. This allows you to permanently set the Data Line according to whether or not you have the Shift Key modification.

>>> SUMMARY OF PRINTING PROCEDURE

To print a document which is in memory, press [P]NP or [P]?NP.

To print a document which is NOT in memory:

1. Press [N]Y to clear memory.
2. [L]oad the desired file into memory.
3. Select print format options or use the Additional Functions Menu to load a previously saved [P]rint/Program Command list.
4. Select either NP (new print) or CP (continue printing).

The Apple Writer][Word Processing Language (WPL) is a very powerful tool that allows you to write your own programs using a special language. If you have understood the manual so far, then you already know much of the language.

The purpose of WPL is to make automated word processing possible. Instead of entering text editor and printing commands from the keyboard, you can write a program to do this for you. In fact, a WPL program is a normal Apple Writer text file that consists of a series of Apple Writer commands. When this program is executed, the computer will perform the Apple Writer commands exactly as if you had typed them in from the keyboard. Hence your computer becomes a player piano that "plays a song" of Apple Writer commands without any hands on the keyboard.

>>> KEY FEATURES

- * WPL can automatically print any number of form letters, each with a different name and address and with other information automatically inserted.
- * WPL can be used to "translate" typewriter shorthand into longhand automatically.
- * WPL can be used to create application programs to perform complex word processing tasks such as report creation, using elements taken from disk files, memory, and user keyboard entries.

>>> WPL COMMAND ORIGINS

WPL commands may originate at the keyboard (the "immediate mode"), a WPL program file, or from within a text file as it is being printed.

WPL programs may contain any Control-character commands that might be issued to the text editor or the [P]rint/Program command menu. However, normal text files may contain only [P]rint/Program commands, not text editor commands. WPL programs have [P]rint/Program commands prefixed by "P". Here are examples:

From the keyboard, press [P], then enter the command:

LM 10 (The default left margin is immediately set to 10)

For this command to be included within a text file, a line at the left margin is entered:

.LM 10 (The left margin is set to 10 during the printing of the text file)

From a WPL program file, it is entered:

PLM 10 (The left margin is set to 10 when the WPL program is executed)

>>> HOW TO WRITE A WPL PROGRAM

Here is the general procedure for writing a WPL program:

1. Using the Text Editor, enter the CTRL commands (as ordinary letters) and parameters that would normally be entered from the keyboard to accomplish some series of Apple Writer][commands. Each command should be on a separate line with a carriage return at the end of each line (like a new paragraph).

NOTE: When writing WPL programs, it is often useful to first "try out" the commands, i.e. manually enter all the Apple Writer commands you wish to include in the program and write down on paper every keystroke you typed.

2. Save this text file to diskette:

```
[S]ave :filename,D1
```

Remember that you only type the CTRL-S; full command names followed by a colon (:) are used in this section of the manual to jog your memory.

NOTE: WPL programs can only be executed from disk.

3. To carry out the program instructions, use the [P]rint/Program "DO" command, entering the file name:

```
[P]rint/Program :DO filename,D1
```

(CTRL-P followed by "DO" command)

NOTE: The "DO" command cannot be executed from the Print/Program Commands Menu.

>>> LESSON 1: TYPICAL WPL PROGRAM

Here is a sample program that prints a number of text files:

```
NY
L FIRSTFILE,D1
PPD 1
PNP
NY
L NEXTFILE,D1
PCP
```

NOTE: Each line of this sample program must begin with one or more spaces at the left margin. Some WPL program lines will include labels beginning at the left margin, as explained below. The indentation of sample programs in this manual is a reminder to enter at least one blank at the beginning of each line.

NOTE: Most sample programs in this manual exist on the master diskette and you can [L]oad them with the editor and experiment with changes.

This program is entered using the text editor just like any normal text file, then saved using the [S]ave command in the normal way (it has been already been saved for you as "EXAMPLE1" on the system diskette). Then, to run the program enter:

```
[P]rint/Program :DO EXAMPLE1,D1
```

(CTRL-P followed by "DO EXAMPLE1,D1")

The [P]rint/Program command "DO" means "carry out the instructions in the WPL program file as if they had been entered from the keyboard".

NOTE: Make sure your printer is connected to slot one and turned on before executing this example program.

Now let's analyze our program. The first rule to remember is that in each line, the first non-blank character following a blank character is interpreted as a CTRL character. Remember that text editor and printing control functions are normally obtained by pressing the CTRL and letter keys. In WPL programs, you only need to enter the letter, without pressing the CTRL Key. WPL will understand what is meant. Therefore, the first line of the program:

```
NY
```

is equivalent to typing CTRL-N, then "Y". Remember that CTRL-N erases memory if followed by "Y". So it means:

```
[N]ew :Y
```

Line two:

```
L FIRSTFILE,D1
```

is interpreted as CTRL-L, followed by a file name and drive number. So it means:

```
[L]oad :FIRSTFILE,D1
```

Line three:

```
PPD 1
```

is a [P]rint/Program command ([P]rint/Program :PD 1) that assigns the destination for printing to slot one.

Line four:

PNP

is a [P]rint/Program :NP command (New Print) which causes the first file to be printed.

Line five:

NY

erases memory before loading the next file.

Line six:

L NEXTFILE,D1

is a [L]oad :NEXTFILE,D1 command which loads the next file to be printed.

Line seven:

PCP

is a [P]rint/Program :CP command (Continue Print) which causes the next file to be printed as a continuation of the first file.

These descriptions show that any functions available as CTRL characters may be entered into a WPL program.

Example: If you wanted to use [F]ind from a WPL program, you would enter:

F/things/stars/

When the program is run, this line is interpreted as:

[F]ind :/things/stars/

Lesson 1 Summary:

<u>WPL Program Line</u>	<u>Result</u>
NY	Clear Memory
L FIRSTFILE,D1	Load FIRSTFILE from Drive 1
PPD 1	Set printer destination to slot one
PNP	Print first file
NY	Clear Memory
L NEXTFILE,D1	Load NEXTFILE from Drive 1
PCP	Continue Printing next file

>>> LESSON 2. REPEATING PARTS OF THE PROGRAM

A program wouldn't be very useful if it could only make sequential commands (as in the example above). It would be more useful if program segments could be repeated. To accomplish this, some special commands are available that may only be used in a WPL program.

Here is a sample program that repeats:

<u>WPL</u>	<u>Program</u>	<u>Line</u>	<u>Result</u>
	NY		Erase memory
	F/**/		Find "nothing" and replace it with "*" i.e. Insert a "*" into memory.
	Y?		Do replacement and stop search (the "?" after the "Y" stops the search)
LOOP	B		Go to beginning of text
	F/**/**/A		Find "*" and replace with "***" (all)
	PIN	Press RETURN to Continue.	Display message and wait for response
	PGO	LOOP	Go to line whose label is "LOOP"

The program line beginning with "LOOP" positions the cursor to the beginning of the text file, equivalent to CTRL-B. The word LOOP is a label since it begins in the left margin. The label is not a WPL command, it just serves as a destination for the line "PGO LOOP". Here are the rules for labels:

1. If a program line needs a label, enter the label at the left margin (before any spaces have been entered). After entering the label, enter one or more spaces before the line command.
2. If there is no need for a label, enter one or more spaces before entering the line command.

NOTE: When WPL encounters a line such as "PGO LOOP", it searches for a line having a matching label at the left margin, and program execution continues at that line.

WORD PROCESSING LANGUAGE (WPL)

The "F/*/**/A" line [F]inds each asterisk in the file and replaces it with two asterisks.

The "PIN Press RETURN to Continue." line displays the message "Press RETURN to Continue." and waits for the operator to press RETURN to continue the program.

The "PGO LOOP" line returns to the line with the label "LOOP" and the process is repeated.

As written, the sample program will fill memory with asterisks. The program may be stopped by pressing the "ESC" key. If memory fills completely, the program will stop automatically due to the overflow condition. Try it: Erase memory (CTRL-N followed by Y) and then enter:

```
[P]rint/Program:DO EXAMPLE2,D1
```

NOTE: The new command in this example is "GO". The program line "PGO LOOP" means "GO to the program line having the label 'LOOP'".

NOTE: It doesn't matter how many spaces lie between the left margin and the line command, or between the label and the line command. This means that spaces can be used to clarify the program structure. Also, there is no limit to label length, although WPL commands should normally be less than 80 characters long for readability. Example:

```
OUTERMOSTLOOP PND
                PSX 10
NEXTLOOP       PSY 10
INNERMOSTLOOP PSZ 10
COUNT        PPR (X),(Y),(Z)
                PSZ -1
                PGO COUNT
                PSY -1
                PGO INNERMOSTLOOP
                PSX -1
                PGO NEXTLOOP
                PGO OUTERMOSTLOOP
```

This program will display three changing numbers in an endless loop. Try it. Enter:

```
[P]rint/Program:DO EXAMPLE3,D1
```

Press ESCape to stop the program.

HINT: Use the [Q] and [I] Tab features described in the Editor section of the manual to assist with formatting program segments as shown above.

>>> LESSON 3. USING NUMBERS IN THE PROGRAM

In the next few lessons, we will show how to use WPL to retrieve a particular mailing address from a master address file and print a form letter using that address.

This method retrieves each address by searching for a number that is changed by the WPL program to refer to each available address.

WPL uses three numbers. As in algebra, letters have been selected that stand for the numbers. The letters are X, Y and Z. These letters are called "variables". This means that they can be changed in value (in the range of 0 to 65535). When they appear in a WPL program, they must be enclosed in parentheses e.g. "(X)".

In WPL we can choose a numeric value for these variables. Here is a program line example:

```
PSX 3
```

Means: [P]rint/Program :SX 3 (Set number "X" to 3).

We may also increment (make larger) or decrement (make smaller):

```
PSX -1    Means "Decrement X by one"
```

```
PSX +4    Means "Increment X by four"
```

EXAMPLE4 Program:

<u>WPL Program Line</u>	<u>Results</u>
PND	Turn off text display
PSX 1	Set variable X to one
LOOP PPR variable now equals (X)	Display message
PSX +1	Increment value of X by one
PGO LOOP	Go to line whose label is LOOP

Try it. Enter:

```
[P]rint/Program :DO EXAMPLE4,D1
```

You will have to press "ESCape" to stop the program.

Notice that "(X)" was replaced by the actual numeric value of "X" as the program ran. Each WPL program reference of the type "(X)" for "(X)" through "(Z)" inclusive will be replaced by the numerical value of the number when the program is run. Here is a sample program line:

PPR Give me (X)!

While running the program, WPL will replace "(X)" with the numerical value of "X" before printing the line (assume that X has a value of 456):

Give me 456!

>>> LESSON 4. USING NUMBERS TO RETRIEVE TEXT FILE SEGMENTS

We can use the variable numbers as an index to a large table of information such as addresses.

Let us say that we have created an address table, with an entry format like this:

```
<1>John Smith
1234 Elm Street
Anytown, U.S.A
54321
```

Each address in the table has its own number. Each address begins with an index number between two angle brackets, e.g. "<1>". Now we can write a program to automatically print form letters:

<u>WPL Program Line</u>	<u>Result</u>
START PSX 1	Set variable X to one
LOOP NY	Erase memory
L FORMLETTER,D1	Load file called FORMLETTER
B	Position cursor to beginning of file
F/(Address)//	Find "(Address)"
Y?	Do replacement and stop search
L ADDR/<(X)>/</N	Load segment from file ADDR
PNP	Print the form letter
PSX +1	Increment value of X by one
PGO LOOP	Go to line whose label is LOOP

The "PSX 1" line sets the numeric variable "X" to 1.

The "NY" line erases memory in preparation for insertion of the form letter.

NOTE: The command "[N]ew" always resets the data line direction to "<".

The "L FORMLETTER,D1" line [L]oads the form letter into the computer's memory. The form letter has the address position marked with "(Address)".

The "B" line positions the cursor to the file [B]eginning and sets the data line direction to ">".

The "F/(Address)//" line [F]inds the "(Address)" character string and prepares to replace it with "nothing" since there is nothing between the second and third slashes in the [F]ind command. The result is that the "(Address)" string is removed from the file. This prevents the "(Address)" string from being printed along with the letter.

The "Y?" line does the replacement, removing the "(Address)" label, then stops the [F]ind feature with the "?" mark. Remember that any character besides RETURN or "Y" will stop the [F]ind feature.

Now the cursor has been placed at the address position. If we insert an address, it will occupy the address position.

The next line [L]oads a file segment, with the following options:

```
L ADDR/<(X)>/</N
```

Remember that "(X)" will be replaced by the numerical value of "X". Therefore, in the first program run, this line will be presented as:

```
L ADDR/<1>/</N
```

This means "[L]oad, from the disk file 'ADDRS', the file segment lying between '<1>' and the following '<' (which begins the next address)". The "N" suffix means do not load the markers themselves ("<1>" and the following "<").

Here is a sample address file that might be sought by our program:

```
<1>John Smith
1234 Elm Street
Anytown, U.S.A
54321
<2>Jane Doe
5678 Main Ave
City, State
98765
```

The [F]ind feature will locate the first address and insert it into the form letter, removing the index number "<1>" as it does so. It is not necessary for the last address to be followed by a "<" because the [L]oad command will load to the end of a file if the ending marker is never found.

The "PNP" line causes the the letter to be printed.

The "PSX +1" line increments the value of variable "X" by one so that it can be used to find the next address.

The "PGO LOOP" line causes execution to proceed with the program line having the label "LOOP" to repeat the entire process with the next address. However this program will repeat indefinitely because it has no way of knowing that it has run out of addresses.

>>> LESSON 5. ERROR FLAGGING AND DETECTION

The previous sample program will print form letters, but it can't tell that it has run out of addresses.

WPL can detect an error in the search of a file of addresses, or, in general, an inability to [F]ind or [L]oad a specified item such as running out of addresses.

When a search fails, a flag is set in WPL that says "an error has been detected". When such an error occurs, the very next program step is skipped, the flag is cleared, and the WPL program proceeds with the second instruction following the one that detected the error.

Here is a version of our sample program that will stop executing when it runs out of addresses:

```

START   PSX 1
LOOP    NY
        L FORMLETTER,D1
        B
        F/(Address)//
        Y?
        L ADDRS/<(X)>/</N
        PGO FOUND          <---- New Program Line
        PGO QUIT           <---- New Program Line
FOUND   PNP
        PSX +1
        PGO LOOP
QUIT    PIN Finished printing

```

The "PGO FOUND" line will cause execution to proceed with the line labeled "FOUND".

The "PGO QUIT" line will cause execution to proceed with the line labeled "QUIT". However the "PGO QUIT" line itself will only be executed if the L ADDRS/<(X)>/</N command is unable to find a segment to load and the "PGO FOUND" step is therefore skipped.

Try it. Enter:

```
[P]rint/Program:DO EXAMPLE5,D1
```

>>> LESSON 6. WPL PROGRAM TERMINATION

Some WPL programs stop by themselves because the program steps are not repeated. Here is an example:

```

PSX 3
PIN "X" Equals (X)

```

This program stops executing because it gets to the end of the list of instructions and there is no "GO" command to repeat any instructions.

Here is a new method to stop a program:

```
START  PSX 1
LOOP   L  ADDR/<(X)>/</N
        PGO FOUND
        PQT                <---- New Program Line
FOUND  PNP
        PSX +1
        PGO LOOP
```

This program stops executing when line 2 fails to find an address with the number "(X)". Then an error flag is set, line 3 is skipped, and line 4 is executed which causes the program to end. The "QT" instruction means "QUIT".

Here is another method to stop a program (EXAMPLE6):

<u>WPL</u>	<u>Program Line</u>	<u>Results</u>
	PND	Turn off text display
	PSX 50	Set variable X to fifty
LOOP	PPR "X" now equals (X)	Display message
	PSX -1	Decrement value of X by one
	PGO LOOP	Go to line labeled LOOP

The instruction "SX -1" (decrement X by one) has been used in line 4. This program will stop when the numeric value of "X" reaches zero because at that time line 5 will be skipped and execution will proceed with line 6 (but there is no line 6 so the program will stop).

NOTE: Whenever a WPL numeric variable is incremented or decremented to a zero value, the "error" flag is set and the next instruction is skipped. If a numeric variable has the largest possible value (65535) and is incremented by one, its value becomes zero and the error flag is set. This is known as "wrap-around".

>>> SUMMARY OF WAYS TO END A WPL PROGRAM

A WPL program will stop if:

1. It runs out of instructions.
2. The operator presses the ESCape key.
3. It executes a "QUIT" command (QT).

>>> LESSON 7. AUTOMATIC TABLE INDEXING

WPL may be used for preparing address tables, as well as printing them. Remember that address tables are simply normal Apple Writer text files that happen to contain lists of addresses with some kind of marker between each address for the search to key on. Let us say that you have developed an address table with the following format:

```
<>John Smith
1234 Elm Street
Anytown, U.S.A
12345
```

Notice that a marked site "<>" has been included, at the position the number will assume. Now we want to automatically number each address in sequence starting with 1. Here is a program that will do this (EXAMPLE7):

	<u>WPL Program Line</u>	<u>Result</u>
	PSX 1	Set variable "X" to 1
	NY	Erase Memory
	L ADDR,D1	[L]oad the address table
	B	Position cursor to the file beginning
LOOP	F/<>/<(X)>/	[F]ind /<>/ and prepare to replace it
	Y?	Do the replacement (if found), and quit search
	PGO FOUND	If the "<>" was found, go to the line having the label "FOUND"
	PGO QUIT	If the "<>" wasn't found, go to the line having the label "QUIT"
FOUND	PSX +1	Increment "X" by one
	PGO LOOP	GO to the line having the label "LOOP" and repeat process
QUIT	PIN Finished	Display message and wait for "RETURN"

When this program is run, you will be able to see the replacement process on the video display. Remember to [S]ave the address file after the indexing is complete.

>>> LESSON 8. PERSONALIZING A LETTER BEYOND THE ADDRESS

It is possible to go beyond simple addressing when personalizing a letter. Here is our example address format:

```
<1>John Smith
1234 Elm Street
Anytown, U.S.A
54321
```

In this example, our form letter will have "(Address)" at the address location and "(Name)" at each location that the first name is wanted. Note that the first name is the segment between "<1>" and a space; the address is the segment between "<1>" and the next "<". Here is a program that will insert the name and address, then go on to insert the first name as many times as it is called for (EXAMPLE8):

<u>WPL Program Line</u>	<u>Result</u>
PSX 1	
LOOP NY	
L FORMLETTER,D1	
B	
F/(Address)//	
Y?	
L ADDR/<(X)>/</N	
PGO GETNAME	
PGO QUIT	
GETNAME F/(Name)//	Find "(Name)" and prepare to replace it with nothing
Y?	Do the replacement and stop the search
PGO FOUND	Go to line labeled "FOUND" if "(Name)" was found
PGO SETUP	Go to line labeled "SETUP" if "(Name)" was not found
FOUND L ADDR/<(X)>/ /N	Load first name from address table into form letter
PGO GETNAME	Go to line labeled "GETNAME"
SETUP PNP	
PSX +1	
PGO LOOP	
QUIT PIN Done at address (X) (Press RETURN)	

This program works because it inserts all between "<(X)>" and "<", exclusive, as an address, and all between "<X)" and " " (space), exclusive, for the first name. To run this program, type:

```
[P]rint/Program :DO EXAMPLE8,D1
```

After seeing this demonstration, [L]oad and examine the files named "ADDRS" (the address table), "FORMLETTER" (the form letter) and "EXAMPLE8" (the WPL program). Careful examination of these files will help show how they work together to create the form letter sequence.

NOTE: If the address file convention described here is used, the system diskette file "AUTOLETTER" may be used for printing. Simply change the printing constants and the destination for your printer before beginning the execution of the WPL program.

>>> WPL SYNTAX ERRORS

If you have difficulties making your WPL programs work, use this checklist to verify that you aren't overlooking anything:

1. Have you [S]aved your file before trying to run it?

```
[S]ave :TESTPRG,D1
[P]rint/Program :DO TESTPRG,D1
```

2. Are ordinary program lines preceded by spaces? Are label lines against the left margin?

```
LOOP      PPR X = (X)
          PSX -1
          PGO LOOP
```

3. Do any of your labels have embedded spaces? Is each label followed by at least one space?

```
FIRST LOOP PPR      (Embedded space in label)

SECONDLOOPPNP      (No space between label "SECONDLOOP" and
command "PNP")
```

4. Is there any difference between labels, such as upper case/lower case conflict?

```
Loop      PSX -1
          PGO loop (Labels are not the same)
```

5. Is a label used more than once? WPL can't decide on its own which label to go to:

```

LOOP      PSX -1
          PGO LOOP      (Label is used twice)
LOOP      PSY -1
          PGO LOOP

```

6. Is the program terminating before it should? If one of the addresses in an address file doesn't have a marker, our EXAMPLE8 program will terminate prematurely:

```

<1>John Smith
1234 Elm Street
Anytown, U.S.A
12345
Jane Doe
5678 Main Ave
City, State
98765
<3>Jim Jones
123 W. 10th
Anywhere, Ohio
57575

```

The address for Jane Doe is missing its marker "<2>". This will cause our WPL program to terminate after printing the form letter for John Smith.

Examining such available WPL programs as the file "AUTOLETTER" on the system diskette may help you to see the structure of WPL.

NOTE: It is not necessary to make WPL entries in UPPER CASE. In the typing of labels, entries need only be consistent. That is, if you mix upper and lower case in a label reference, the label itself must be identical. For example:

```

FuNnY1Abel PSX +1
          PGO FuNnY1Abel

```

NOTE: See the "WPL Execution Errors" section of the manual for an explanation of some error messages that may be displayed while a WPL program is running.

>>> ADVANCED WPL TECHNIQUES AND FEATURES

WPL has a number of powerful features beyond those described earlier, such as string variable assignment and comparison, subroutines, and user entry to a running WPL program.

First, it is important to review the role played by the "Error flag". When an error condition is found, the next WPL instruction is always skipped. Example:

```
L FILE /WORD./
PGO NEXTPART
```

In this example, if the specified file segment isn't found, the "PGO" instruction won't be executed, because an error is present.

>>> STRINGS IN WPL

Strings (groups of characters) may be entered into WPL. There are four string variables (\$A,\$B,\$C,\$D) to which strings may be assigned. These strings may then be printed, added to one another, used for searches, and so forth.

Here is an example of string assignment:

```
PAS Here is the string =$A
```

"PAS" means "Assign the characters that follow to the specified string variable". The blank character after the "PAS" and the blank character before the equal sign are included as part of the string.

Here is a string assignment from the keyboard, rather from the program itself:

```
PIN Enter your name =$A
```

The "PIN" instruction, otherwise used to prompt the operator, is used in this example to get an entry from the keyboard and place it in a string variable. Whatever the user types in will be placed into the \$A variable.

To print a string variable, simply place it in a printed line or anywhere that its contents may be useful:

```
PIN Is your name $A ? =$B
```

This line prints the string value of \$A as part of the prompt, then accepts a keyboard entry for \$B. There are other ways that a string may be used:

```
F/$A/$B/A          (Replace all occurrences of the
                    string value of $A with the
                    string value of $B)
```

```
PIN Enter your age : =$A (get a number)
```

```
PSX $A             (Assign the number in $A to (X))
```

```
PAS (X) =$A        (Assign the number (X) to $A)
```

```
PIN Select (A - F) =$A (A menu example)
PCS/$A/A/           (If $A = "A" then no error)
                    (Therefore GO to CHOICEA)
PGO CHOICEA
PCS/$A/B/           (If $A = "B" then no error)
                    (Therefore GO to CHOICEB)
PGO CHOICEB         (and so forth)
```

The command "PCS" means "Compare Strings". If the strings are not equal, the the error flag is set. Remember that if the error flag is set, the next WPL instruction is skipped.

```
PAS $A$B =$C ($C equals the concatenation
              of $A and $B)
PSX (Y)       (number (X) is set to (Y))
```

In WPL string variable usage, an assignment looks like this: "=\$A". There can be no intervening spaces. To use the contents of a string variable, enter "\$A". Again, no spaces.

NOTE: The four string variables (\$A,\$B,\$C,\$D) may contain up to 64 characters each. A bell will ring if an entry is longer than this, but the first 64 characters will be used and the program will continue.

String variables may also have their values loaded from a file:

```
PLS TESTFILE/The./ =$A
```

A phrase beginning with "The" and ending with a "." will be loaded from the file "TESTFILE" and placed into the string variable \$A. This load will NOT affect the current file in memory. Care must be exercised so that the string size is not exceeded.

>>> WPL SUBROUTINES

A "Subroutine" is a set of instructions that may be accessed repeatedly from anywhere in the program. It is a special "side trip" from which the program eventually returns. Subroutines can be effectively used to reduce the number of instructions in a WPL program by reusing the same instructions multiple times. Here is an example:

```
FIRST    PSR DEVICE
NEXT     PIN =$B
        .
        .
        PGO QUIT
DEVICE   PSX -1
        PAS (X) =$A
        PRT
QUIT     PPR End of Program
```

The command "PSR" Means "carry out the instructions beginning at the label 'DEVICE' and return here when the instruction 'PRT' is encountered". The program would carry out the "PSR" instruction, go to the subroutine at "DEVICE", then return to the instruction following the PSR instruction ("NEXT" in this case) and continue from there.

The subroutine instructions make it possible to create programs, in which essential and frequently-used functions are written in only one location, then called from anywhere in the program. Subroutines may call subroutines, up to 32 levels, and the program will find its way back.

NOTE: It is a good idea to place subroutines at the beginning of the program for maximum operational speed. A program structure like this is best:

```
START    PGO MAINLINE
SUBA     (code for subroutine A)
        PRT
SUBB     (code for subroutine B)
        PRT
        (etc.)
MAINLINE (begin main program)
        .
        .
        PSR SUBA
        .
        .
        PSR SUBB
```

WORD PROCESSING LANGUAGE (WPL)

>>> SCREEN DISPLAY DURING WPL PROGRAMS

It is frequently desirable to display menus, results, and questions during the running of a WPL program. There are some techniques to improve the appearance of the display.

In a menu program it is desirable to clear the screen before printing the menu. It is also a good idea to suppress the text editor display while the menu is being displayed. Here is an example menu program (EXAMPLE9).

<u>WPL Program Line</u>	<u>Result</u>
START PND	Turn off text display

(Next we clear the screen by sending a CTRL-L to the screen. The CTRL-L will appear as an inverse "L" while the program is being written.)

```

PPR          (CTRL-V, CTRL-L, CTRL-V to clear screen)
PPR **** MENU PROGRAM ****
PPR
PPR          Display blank line
PPR Options:
PPR
PPR (A) Find cases of a word
PPR (B) Count words in file
PPR (C) Reorder a file
PPR (D) Enter text editor
PPR (Q) Quit
PPR
PIN Select (A - D) : =$A
PCS/$A/Q/    Is $A = "Q"?
PQT         Yes. Quit
PCS/$A/A/    Is $A = "A"?
PGO FIND     Yes. go to FIND routine
             and so forth.
.
.
FIND        PIN Here is the FIND routine
             PGO START
    
```

Try it. Enter:

[P]rint/Program :DO EXAMPLE9,D1

NOTE: In the preceding program, each "PGO" instruction can be replaced by a "PDO" instruction, to run individual WPL files rather than simply branching within one file. This allows the ability to "chain" from one program to another.

>>> WPL PROGRAM COMMENTS

Comments (words that don't cause any action) may be entered into WPL programs for documentation. Here is an example:

```
P This is my comment.
```

This method doesn't interfere with program operation because the command "P" signifies a [P]rogram command entry, but there are no valid [P]rogram commands beginning with a space. Therefore the comment causes no action to be taken when the program is running. Comments must be less than 128 characters in length.

>>> WPL PROGRAM LENGTH

WPL programs may be up to 2048 characters in length. The "Len" specifier in the text editor Data Line can be used to monitor the program length as it is written. If a program exceeds this length, it may be broken into segments in this way:

At the end of the first segment, place this line:

```
ENDOFONE   PDO SEGMENTTWO
```

Now write a second segment called SEGMENTTWO. This process may be repeated as often as necessary. Any of these additional segments may then "PDO SEGMENTONE" to return to the first segment.

If an attempt is made to run a program greater than 2048 characters in length, an error message will be generated (see "WPL Error Messages" below).

NOTE: If the WPL program is controlling printing, and the printed file uses footnotes, then the maximum WPL file size is 1024 characters.

>>> CONTROLLING TEXT EDITOR DISPLAY FROM WPL PROGRAMS

While developing a program, seeing the text editor display is beneficial in order to see the effect of the WPL commands on the current text file. After the program has been written and verified, the text editor display should be suppressed unless absolutely necessary. Suppressing the display:

PND (No text editor display)

makes WPL programs run about 5 times faster. WPL printing commands are still carried out; only the text file display is suppressed by this command.

>>> WPL LIBRARY CREATION

Because there are no line numbers in WPL to limit programming flexibility, a set of proven program routines may be developed, and saved to diskette. Then this library may be used to create WPL programs rapidly. Whenever a function is needed, simply [L]oad it and insert it into the developing program.

>>> WPL EXECUTION ERRORS

While a WPL program is being run, some execution errors may arise. These errors are displayed as:

WPL Error (Message)

Here are the error messages and their meanings:

"WPL Error: Label not found --> xxxxx (Press RETURN)"

This means that the label "xxxxx" was called for but not found. Here is an example that would give this error message:

```
START      PND
           .
           .
           .
           PGO start
```

The error is that one of the label references is capitalized, the other is not.

"WPL Error: 'RT' without 'SR' (Press RETURN)"

This error is caused by attempting to return from a subroutine that was never called. The "PRT" command may only be used subsequent to a "PSR" instruction that passes program control to the subroutine. Example:

```
START    PND
          PSX -1
          PRT
```

If this were the program beginning, then there is nowhere for the "PRT" instruction to return to, since this program segment wasn't called as a subroutine.

"WPL Error: Program > 2048 chars (Press RETURN)"

This error arises if a program greater than 2048 characters in length is run. See the section entitled "WPL Program Length" for methods to divide a WPL program into segments.

"WPL Error: More than 32 'SR' (Press RETURN)"

This error means that more than 32 "PSR" go-to-subroutine instructions have been encountered without any "PRT" return-from-subroutines. This usually arises from programming errors, such as:

```
START    PSR START
```

"WPL Error: Footnote overflow (Press RETURN)"

This error is caused by the presence of more than 1024 characters of footnotes in a single page. Space for 1024 characters of footnotes is provided, this WPL Error arises if the available space is completely filled by the footnotes for a single printed page.

>>> WPL EXAMPLE PROGRAMS

Following are some example WPL programs that show typical applications.

Word Counter

Counts occurrences of a word or string in a file. This program exists on the system diskette as COUNTER.

```

START  PND
        PPR      (CTRL-V, CTRL-L, CTRL-V to clear screen)
        PPR      Word Counter
        PIN      Enter Disk File Name:=$A
        PCS/$A//
        PGO      QUIT
NAMEOK NY
        D
        L $A
        PGO      WORDS
        PGO      START
WORDS  PIN      Enter word(s):=$B
        PCS/$B//
        PGO      START
        PSX      0
        B
LOOP   F/$B/$B/
        Y?
        PGO      SETUP
        PGO      TOTAL
SETUP  PSX +1
        PGO      LOOP
TOTAL  PIN      Total number of "$B" in file "$A" = (X) (Press RETURN)
        PGO      WORDS
QUIT  PIN      End of Word Count      (Press RETURN)

```

Autoletter

Prints any number of form letters, using an address file named "ADDRES" and a formletter named "FORMLETTER". This program is described at length earlier in the manual. Simple changes to this program will make possible the entry of keyboard information into the letter, disk-based text blocks, and so forth. This program exists on the system diskette as AUTOLETTER.

```

START  PSX 1
LOOP   NY
       L FORMLETTER,D1
       B
       F/(Address)//
       Y?
       L ADDRES/<(X)>/</N
       PGO FOUND
       PGO QUIT
FOUND  PLS ADDRES/<(X)>/ /N=$A
       B
       F/(Name)/$A/A
       PNP
       PSX +1
       PGO LOOP
QUIT   PIN           Done at address (X) (press RETURN)
       NY

```

File Mover

This program automatically moves files from one disk to another. Copying the entire disk is the preferred method, but this is not always possible. This program exists on the system diskette as MOVER.

```

START      PND
           PPR Enter the Source disk drive
           PPR
SOURCE     PIN (Example entry: "D1"): =$A
           PCS/$A//
           PGO SOURCE
           PPR
DEST      PIN Enter the Destination drive: =$B
           PCS/$B//
           PGO DEST
MOVER     PPR
           PIN Enter File Name (Q = Quit): =$C
           PCS/$C/Q/
           PGO QUIT
           PCS/$C/q/
           PGO QUIT
           PCS/$C//
           PGO MOVER
           NY
NOTICE    L $C,$A
           S $C,$B
           PGO MOVER
QUIT     PIN End of File Mover      (Press RETURN)
           NY

```

Notice about this program that the comma between "\$C" and "\$A" in the line labeled "NOTICE" is not a printing format command, instead it is an ordinary character that is printed. This results in a format of:

```
"filename,D1"
```

Automatic Replacement

This program makes a string replacement across many files. Thus untrained users can operate it without fully understanding Apple Writer][. This program exists on the system diskette as REPLACE.

```

START      PND
           PPR **** Automatic String Replacement ****
           PPR
SOURCE     PIN Enter the string to be sought : =$A
           PCS/$A//
           PGO SOURCE
           PPR
REPLCMT    PIN Enter the replacement : =$B
           PCS/$B//
           PGO REPLCMT
           PPR
NEXTFIL    PIN Enter file name (Quit = Q) : =$C
           PCS/$C//
           PGO NEXTFIL
           PCS/$C/Q/
           PGO QUIT
           PCS/$C/q/
           PGO QUIT
           NY
           L $C
           PSX 0
           B
LOOP       F/$A/$B/
           Y?
           PGO SETUP
           PGO SAVE
SETUP      PSX +1
           PGO LOOP
SAVE       S $C
           PPR
           PIN (X) cases of "$A" replaced by "$B" in file "$C"
           (Press RETURN)
           PGO NEXTFIL
QUIT       PPR
           PIN End of Automatic Replacement (Press RETURN)
           PYD

```

Automatic Print

Prints any number of different documents after prompting the user for the names of the files. This program exists on the master diskette as AUTOPRINT.

```

START  NY
       PND
       PPR          (CTRL-V, CTRL-L, CTRL-V to clear screen)
       PPR **** AUTOMATIC PRINT ****
NAME   PIN Enter name of file to be printed: =$A
       PCS?$A??
       PGO QUIT
       E
       D
       F<< NY><
       Y?
       F<< L $A><
       Y?
       F<< PNP><
       Y?
       PPR          (CTRL-V, CTRL-L, CTRL-V to clear screen)
       PPR **** AUTOMATIC PRINT ****
QUIT   PGO NAME
       F<< NY><
       Y?
       F<< PIN End of Automatic Print.      (Press RETURN)><
       Y?
       S PRINTIT,D1
       NY
       PYD
       PDO PRINTIT

```

NOTE: After executing this program, try loading the file called PRINTIT. As you can see, AUTOPRINT has created another WPL program (PRINTIT), and then executed it.

APPENDIX A

WPL COMMANDS SUMMARY

<u>Command</u>	<u>Action</u>
AS	Assign text to specified string
CS	Compare strings; skip command
LS	Load string
DO	Perform specified WPL file
QT	Quit WPL, return to editor
GO	Go to specified label in WPL file
SR	Subroutine call
RT	Return from subroutine
ND	Disable text editor display
YD	Enable text editor display
PR	Print a line (including strings if specified)
IN	Display message and prompt user
NP	Begin printing text in memory
CP	Continue printing text in memory
SX	Set value of variable X
SY	Set value of variable Y
SZ	Set value of variable Z

APPENDIX B

SPECIAL CHARACTERS GLOSSARY FILE

The keystroke sequences necessary to obtain the special characters which are not shown on your keyboard (such as {, }, ~ and so forth) can be difficult to remember and awkward to enter. In addition, many printers have special features which can be enabled by sending a special character or sequence of characters to the printer. As discussed in the "Superscripts and Subscripts" section of the manual, the Qume Sprint 5 printer uses the sequences ESC SHIFT-U and ESC SHIFT-D for subscripting and superscripting text. Apple Writer's [G]lossary definitions feature can be used to easily enter these characters into your text file.

Your Apple Writer][master diskette contains a glossary text file named "SPECIAL" which has terms defining special characters and the keystroke sequence necessary for sub- and superscripting. If the file "SPECIAL" is loaded using [Q]5 (Load [G]lossary file), you can type [G]i to embed the reverse half-line feed instruction and [G]m to embed the forward half-line feed. For example, if you wanted the number 3 to appear in superscript position in your printed text you would enter:

[G]i3 [G]m

To print characters in subscript position, just reverse the procedure:

[G]m3 [G]i

The table below lists the [G] character defining each special character.

<u>[G] Character</u>	<u>Special Character Or Keystroke Sequence</u>
b	\
c	^
t	~
u	
v	
<	[
>]
({
)	}
i	[V] ESC D [V]
m	[V] ESC U [V]

APPENDIX C

LOADING THE DEFAULT TAB SETTINGS AND [P]RINT/PROGRAM VALUES

When you boot the Apple Writer][master diskette, you see the message, "Press RETURN to begin:" (If you don't have an 80-column board, you must first respond to the question, "Do you have lower case display (Y/N)?", then you see the "Press RETURN to begin" prompt.) When you press RETURN, the Apple Writer][system files for tab stops and [P]rint/Program Command values are loaded into memory.

If you wish to load alternate default values, but want to retain the default values contained on the master diskette, follow this procedure:

1. Change the tab settings and [P]rint/Program Command values to the desired default values as explained in the manual (see the sections on "Tabulation" and "Changing Format Values Using CTRL-P). Also included in the default values is the current Shift Key setting on the Data Line.
2. [S]ave these new values using the Additional Functions Menu ([Q]) on an initialized diskette (NOT the Apple Writer][master diskette) under the file name SYS .
3. Boot Apple Writer][. DON'T "press RETURN to begin" yet.
4. Take the master diskette out of drive one and insert the data diskette which contains the alternate SYS default values in drive one.
5. Now press RETURN.

The alternate default values will be loaded into memory.

NOTE: THIS PROCEDURE WILL WORK ONLY IF YOU [S]AVE THE NEW VALUES USING [Q]2 AND [Q]4 UNDER THE FILE NAME SYS.

APPENDIX D

USING APPLE WRITER][WITH GOODSPELL

Apple Writer][can be used in conjunction with the GOODSPELL spelling checker program also available through Special Delivery Software. To perform a spelling check on an Apple Writer][file, follow these steps:

1. [L]oad the file to be checked and position the cursor to the beginning of the file [B].
2. Press [C] until the cursor displays a U.
3. [S]ave the file onto your data diskette using a temporary file name such as TEXT.CHECKSPELL. The file name MUST begin with "TEXT.".
4. Now [S]ave the file again using [S]ave :=. Notice that the entire file is in upper case. This is necessary because GOODSPELL only understands Apple Writer][words that are in upper case.
5. Boot your GOODSPELL diskette. When prompted for the Apple Writer file name, insert your data diskette and enter just the part of the name following the "TEXT." prefix. (CHECKSPELL in our example).
6. Make note of the spelling errors and then reboot Apple Writer][in order to correct the spelling errors.

NOTE: THIS PROCEDURE WILL WORK ONLY IF YOU [S]AVE
A FILE THAT IS TOTALLY UPPER CASE AND WHOSE
FILE NAME BEGINS WITH "TEXT.".

NOTES

Apple Writer][Reference Card

Commands using the CTRL key are shown as [].
 All commands may be entered in upper or lower case.

EDITING

<u>Command</u>	<u>Effect</u>
ESC ESC	Cursor move mode:
I	up one line
J	left one character
K	right one character
M	down one line
E	up 12 lines
S	left 24 characters or one word
D	right 24 characters or one word
X	down 12 lines
<--	Deletes character(s)
-->	Retrieves character(s) deleted by <--
[B]	Moves cursor to file beginning; sets Data Line arrow to >
[C]	Automatic case change; cursor displays L or U
[D]	Changes direction of Data Line arrow
[E]	Moves cursor to file end; sets data line arrow to <
[F]/{text}/{replacement}/	Finds and deletes {text} and inserts {replacement}
[G]?{character}{text}	Defines glossary term
[G]{character}	Inserts text previously defined by {character}
[G]*	Deletes all glossary definitions in memory
[I]	Tabs to position set by [T]
[K]	Alpha lock; displays U on Data Line
[L] FILENAME,d#	Loads FILENAME into memory from diskette in drive #
[L]#/marker1/marker2/	Copies specified text segment from memory to cursor position
[L] FILENAME,d#\	Displays FILENAME on screen
[L]?,d#	Lists file names on diskette in drive #
[N]	Erases memory
[O]	DOS Commands Menu
[P]?	[P]rint/Program Commands Menu
[Q]	Additional Functions Menu
[R]	Replace edit mode
[S] FILENAME,d#	Saves memory contents under FILENAME on diskette in drive #
[T]	Tab set/clear/purge
[V]	Accepts CTRL characters as text entries; V appears on Data Line
[W]	Word delete or insert
[X]	Paragraph delete or insert
[Y]	Split screen enable/disable
[Z]	Enable/disable word wrap-around; Z on Data Line when enabled

Apple Writer][Reference Card

PRINTING

<u>Command</u>	<u>Effect</u>	
[P]NP	Prints text currently in memory	
[P]CP	Continues printing	
[P]?	Displays print format options	Default Value:
LM{number}	Sets left margin to {number}	9
PM{number}	Indents/Outdents paragraph margin to {number}; value relative to "LM"	0
RM{number}	Sets right margin to {number}	79
TM{number}	Sets top margin to {number}	1
BM{number}	Sets bottom margin to {number}	1
PN{number}	Sets {number} of first page printed	1
PL{number}	Sets {number} of printed lines per page	58
PI{number}	Sets page interval (form length) to {number}	66
LI{number}	Sets {number} of spaces between lines	0
SP{0 or 1}	Selects continuous sheet feed {0} or single page {1} printing	0
PD{number}	Selects print device destination slot	1
CR{0 or 1}	Selects {1}/deselects {0} line feed signal to printer	0
LJ	Left justify	
FJ	Fill justify	LJ
CJ	Center justify	
RJ	Right justify	
TL	Specify top line headings	blank
BL	Specify bottom line headings	blank

The Print/Program commands and the commands which follow can be embedded in your text by preceding them with a carriage return and a period. Each text-embedded command must be entered on a separate line.

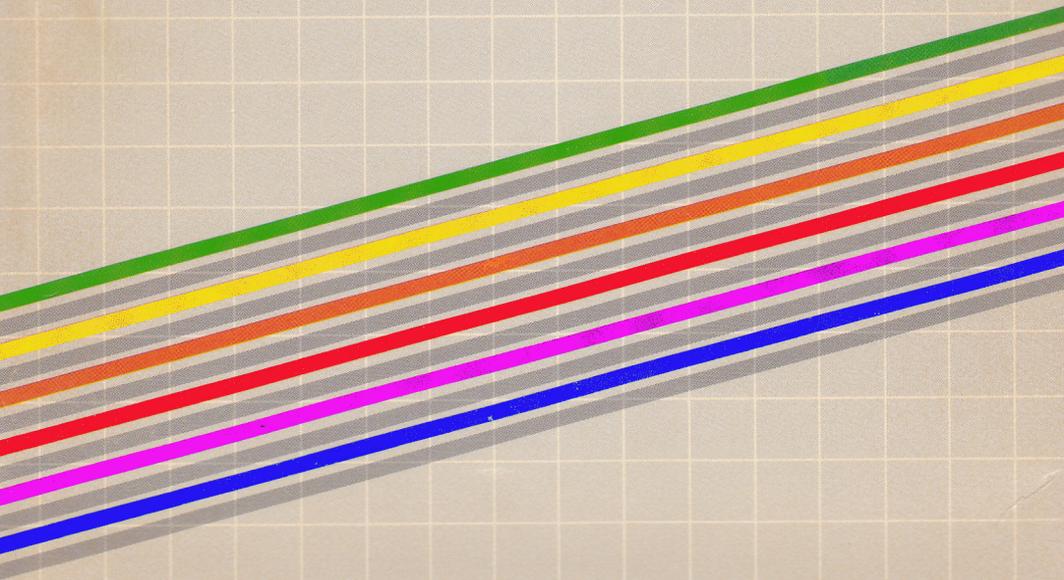
.FF	Form feed
.FF{number}	Conditional form feed if less than {number} lines left on page being printed
.IN{text}	Stops printing and displays {text} on monitor
.EP{0 or 1}	Suppresses {0} or enables {1} printing



APPLE WRITER II
C2B0010 BACKUP

© 1981 Lulus

680-0116



 **apple computer inc.**

© 1980 by APPLE COMPUTER INC.