

REMOTE CONTROL APPLE WRITER

N

WORKING SMARTER

Normally, it is impossible to print Apple Writer files and show all the embedded commands and control characters. Now you can document all those hidden commands, with this useful utility.

For the casual user, Apple Writer II is an easy-to-use word processor. For the more sophisticated user, however, a powerful facility can turn this reliable engine into a virtual dynamo. This facility is known as WPL, which stands for Word Processing Language. By using standard programming techniques, it is possible to harness the power of the Apple Writer word processor to maintain small databases, merge form letters with mailing lists, automate printing and prepare documents for a variety of printers.

There is one problem with WPL files, though. How do you print them? WPL files

frequently contain control characters and other special characters which interfere with normal printing. WPL.PRINT (Listing 1) is a WPL program that will replace the troublesome characters with printable symbols. This allows you to print your WPL programs in an easily readable form. In addition, it generates line numbers for the printout, making it even easier to read. It also will print out regular Apple Writer files, showing all the embedded commands and control characters.

For greater clarity, the following notation has been adopted in this article:

1. Control characters are identified by square brackets and highlighting, so that **[A]** corresponds to Control-A. (This is consistent with the Apple Writer documentation.)
2. Special keys or characters that cannot be printed are identified by braces and highlighting, so that **{esc}** corresponds to the Escape key.

USING THE PROGRAM

To print an Apple Writer II file with the embedded commands and control characters, you should type:

```
[P]DO WPL.PRINT
```

The program prompts you for the file name, disk drive, date and option. It clears memory, loads the file, performs the necessary string substitutions, and produces the appropriate printout. Then it gives you the opportunity to select a new file or to quit.

Example 1 is a printout showing the effects of a variety of printer commands available on the Epson FX-80. When the file used to generate Example 1 is printed using the All option of WPL.PRINT, it appears as shown in Example 2. If you have an FX-80, you can use Example 2 to enter the file.

All of the control and special characters in Example 2 are displayed in boldface print. The carriage returns at the ends of the lines are represented by a custom carriage

return character. If your printer has bit map or programmable character capability, you can include such a symbol. See the instructions in the Entering the Programs section.

WPL.PRINT is also capable of printing any normal WPL program (see Listing 1). To accomplish this, simply select the WPL option, rather than All, when prompted by the program. In order to print WPL.PRINT itself, you must insert a space between (and < in lines 118 and 119 and between > and) in lines 120-121 to prevent Apple Writer

Where there are spaces at the beginning of a line, you just need to enter one space. The extra spaces are for neat formatting. Lines with labels at the beginning should not be preceded by a space.

To enter control codes and Escape, indicated by highlighted letters in square brackets and {esc} in the listing, type [V] to activate control character insertion mode. When this mode is active, the letter V appears at the left end of Apple Writer's status bar, and the control characters show up

1. Replace "L \$A,\$SB" in line 76 with "LSA".
2. Add a line before line 144: " PTL"
3. Delete lines 54-60.

Other Printers

To modify Listing 1 to work with other printers, you'll need to change the commands to turn boldface type on and off. Consult your printer manual for the proper codes, which may also be listed as controlling emphasized or double-strike modes. Substitute the boldface-on command for {esc}E and the boldface-off command for {esc}F in lines 92, 122 and 134 and every other line in the range 98-116.

Although lines 98-116 contain substitutions for every commonly used control code, you may want to add more. If you have Apple Writer II (but not IIe or IIc), you can add [Q], which is used in a number of Epson commands. To add [Q], insert the following two lines before line 100:

```
F/[Q]/[esc]E[Q][esc]F/A
PPR ctrl-QASB
```

Use the same format to add substitutions for other control codes.

Adding a Carriage Return Symbol

Instead of the {cr} used in Listing 1, you may want to use your printer's bit map or programmed character capability to generate your own custom symbol. Printers vary widely in their commands and capabilities in these areas. Three examples are given below.

Epson MX-80 With Grafrax Plus — The method described here works only with the Apple Writer II version. See your Apple Writer manual for instructions on how to enter the underline character. First, change line 134 of Listing 1 to read as follows:

```
F<><[esc]L[G][@]0HL[T][T][K]><A
```

Then add the lines shown above to Listing 1 for [Q] substitution.

Epson FX-80 — Listing 2 installs a programmed character before Apple Writer is booted. With DOS 3.3 booted, key in the Applesoft listing and save it with the command:

```
SAVE FX80.PC
```

In addition, line 134 of Listing 1 must be changed:

```
F<><[esc]E[Q][esc]F><A
```

ImageWriter — In order to use custom characters on the ImageWriter, they must first be loaded into the printer's memory. This cannot be done from within Apple Writer. The Applesoft program in Listing 3 will load a custom character into the ImageWriter (see Figure 3). (An assembly listing of the machine language portion is

FX80.CHARS ... 3/24/86

This is an illustration of some of the character fonts produced by the Epson FX-80 printer:

```
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
ABCDEFGHIJKLMNPOQRSTUVWXYZ
```

```
abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz
abcdefghijklmnopqrstuvwxyz
```

The quick brown fox jumped over the lazy dog in proportional mode.

```
0123456789
0123456789
0123456789
0123456789
```

```
!"#$%&'()*+,-./:;<=>?@ [ ] ^ _ ` { | } ~
```

```
the backslash
is missing
because it
produces
underscores ----->
```

```
!"#$%&'()*+,-./:;<=>?@ [ ] ^ _ ` { | } ~
!"#$%&'()*+,-./:;<=>?@ [ ] ^ _ ` { | } ~
```

EXAMPLE 1: Demonstration of FX-80 Type Style Commands

from creating a footnote as it normally would when encountering these characters. This modified version should be saved as a separate file. It will not work properly with the extra spaces in it. Use the unmodified version to list the modified one.

ENTERING THE PROGRAMS

For Epson Printers

Listing 1 is a version of WPL.PRINT that works with the II, IIe and IIc versions of Apple Writer and Epson printers. (See below for modifications that may be necessary or desirable for your printer.) To enter the listing, boot Apple Writer, clear memory with [N], and enter the lines as shown (pressing Return after each line). Be sure to omit the line numbers and ensuing colons. These numbers are used for reference only.

Where there are spaces at the beginning of a line, you just need to enter one space.

as inverse characters on the screen. In addition, cursor movements, the Delete key and most Apple Writer commands will not work. To turn off this mode, press [V] again. Apple II Plus users should consult the Apple Writer manual for directions for entering characters such as the backslash and curly brackets.

Save the program by pressing [S] and typing WPL.PRINT in response to the file name prompt.

ProDOS

In the ProDOS version of Apple Writer, spaces and drive numbers are not permitted when specifying files. Also, there is a bug in Apple Writer that causes the header line to persist, even though no header line is shown in the printer command menu. The following changes are necessary:

shown, for informational purposes only, in Listing 4.) With DOS 3.3 booted, key in Listing 3 and save it with the command:

SAVE IMAGEWRITER.PC

Be sure to run it *before* booting Apple Writer, and don't turn off the ImageWriter or the character will be lost.

In addition to Listing 3, several changes are required to Listing 1. Substitute the boldface-on and boldface-off commands (`{esc}!` and `{esc}''` respectively) as described under Other Printers. Finally, change line 134 to read:

```
F<<{esc}!{esc}•8{esc}$ {esc}''><A
```

EMBEDDED COMMANDS

If the first character following a carriage return is a period, then Apple Writer II interprets everything up to the next carriage return as an embedded command. Although an embedded command may affect the format of the printing, the command itself does not get printed. Since one of the options of WPL.PRINT is to "print it all," it is necessary to trick Apple Writer into thinking that there are no embedded commands in the file.

To accomplish this, WPL.PRINT inserts `[Q]`, which is a nonprinting character, before every period following a carriage return (line 130). However, if the file begins with an embedded command, there is no carriage return preceding it. PRINT.FILE temporarily

inserts a carriage return at the beginning of the file to handle this situation (lines 77-79).

SPECIAL CHARACTERS

A similar technique is used to handle the footnote delimiters. Footnotes are delimited by the following pairs of characters: (< and >). Inserting `[Q]` allows them to be printed as ordinary text (lines 118-121). On the other hand, Escape, backslash and control characters are merely replaced by suitable character strings, which are then printed in boldface (lines 98-117 and 122-123).

CARRIAGE RETURNS

Because of their use in embedded commands and paragraphs, it is helpful to have a printout of an Apple Writer II file with the carriage returns identified. Listing 1 replaces all carriage returns with the character string `{cr}`. However, I found it more challenging to produce a unique carriage return symbol.

Epson MX-80 With Grafrax Plus

One way to print bit image graphics is to send the Epson MX-80 the following sequence of characters:

```
{esc} L n [G] v1 v2...vn
```

where *n* is the number of dot columns to be printed and *v1...vn* are the *n* values that

will cause the appropriate pins to fire in each column. The values *v1...vn* are sent as characters and are determined by the following steps:

1. Lay out the desired pattern on a grid to identify the pins that need to be fired.
2. Determine the numerical value that corresponds to each pin and the values in each column.
3. Convert each value into the appropriate ASCII character.

In Figure 1 the graphics symbol for the carriage return has been laid out on a grid. By adding the values shown to the right for each pin that is to be fired, a total value is determined for each column. Use a table to convert the hex values into ASCII characters. The character sequence required to produce this graphics symbol is:

```
{esc} L [G] [0] 0 H H _ [T] [T]  
[K]
```

which is precisely what appears in the alternate version for line 134 of Listing 1. Note that the IIe and IIc versions of Apple Writer cannot include the `[@]` character in their files.

Epson FX-80

The Epson FX-80 has the ability to use programmed characters. Because it's impossible to enter the ASCII null character `[@]` in files with the IIe and IIc versions of Apple Writer, it was necessary to write a separate BASIC program that performs all the steps of programming the carriage return character before Apple Writer is booted.

The coding for FX-80 programmed characters (Figure 2) is similar to that described above for bit mapped characters on the MX-80. The FX-80 can fire pins in 11 columns. The intermediate five columns help to make smoother edges on the characters. The only rule is that the same pin cannot be fired in consecutive columns. The pattern for the carriage return character is mapped out in Figure 2 with the column totals below.

The Applesoft program (Listing 2) switches the FX-80 to look for characters in RAM, copies the ROM characters to RAM, defines the programmed character and executes a command that allows the FX-80 to print characters assigned to ASCII values normally reserved for non-printing control characters. The carriage return symbol replaces ASCII character 21 `[Q]`. The format for the character definition command is:

```
{esc} & [G] c1 c2 attr1 b(1,1) b  
(1,2)...b(1,11) attrn b(n,1)...b  
(n,11)
```

where *c1* and *c2* are the starting and ending characters in the range to be replaced. The attribute byte (*attr1* or *attrn*) is a specially coded value that determines how each character is printed in the FX-80's proportional mode and whether the character is printed

FX80.CHARS ... 3/24/86

Page 1

```
.tlq  
.blq  
.lml0q  
.rns0q  
.llq  
This is an illustration of some of the character fonts produced by the  
Epson FX-80 printer:q  
q  
.lml5q  
ABCDEFGHIJKLMN0PQRSTUvwxyzq  
[Q]ABCDEFGHIJKLMN0PQRSTUvwxyz[CR]q  
{esc}4ABCDEFGHIJKLMN0PQRSTUvwxyz{esc}5q  
{esc}EABCDEFGHIJKLMN0PQRSTUvwxyz{esc}Fq  
{esc}-1ABCDEFGHIJKLMN0PQRSTUvwxyz{esc}-0q  
{esc}p1ABCDEFGHIJKLMN0PQRSTUvwxyz{esc}p0q  
{esc}MABCDEFGHIJKLMN0PQRSTUvwxyz{esc}Pq  
[N]ABCDEFGHIJKLMN0PQRSTUvwxyz[T]q  
q  
abcdefghijklnopqrstuvmxyzq  
[Q]abcdefghijklnopqrstuvmxyz[CR]q  
{esc}4abcdefghijklnopqrstuvmxyz{esc}5q  
{esc}Eabcdefghijklnopqrstuvmxyz{esc}Fq  
q  
{esc}p1The quick brown fox jumped over the lazy dog in proportional  
mode.{esc}p0q  
q  
0123456789q  
{esc}S101234{esc}S056789{esc}Tq  
[Q]0123456789[CR]q  
[Q]{esc}S00123456789{esc}T[CR]q  
q  
!"#$%&'()*+,-./:;<=>?@ (backslash) [Q] _ (11) (backslash)q  
q  
the backslash is missing because it produces underscores -----q  
q  
{esc}4!"#$%&'()*+,-./:;<=>?@ (backslash) [Q] _ (11) (backslash)q  
{esc}E!"#$%&'()*+,-./:;<=>?@ (backslash) [Q] _ (11) (backslash)q  
.ff
```

EXAMPLE 2: WPL.PRINT Printout of Example 1

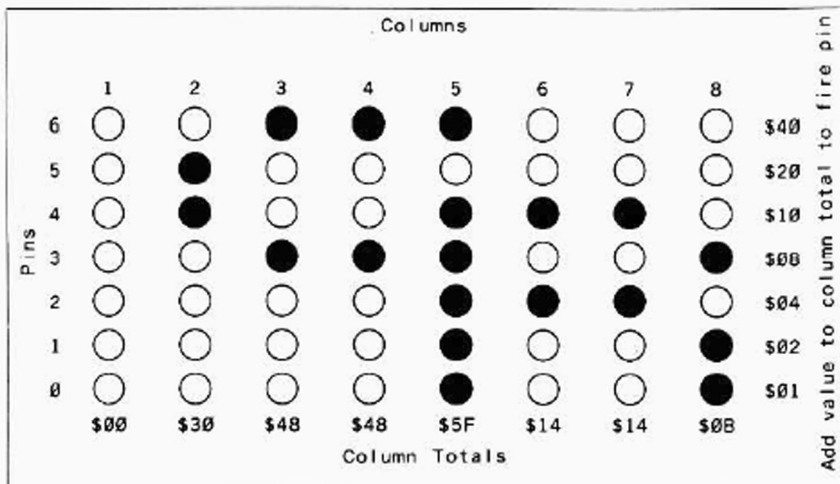


FIGURE 1: Bit-Mapped Character for Epson MX-80 with Grafrax Plus

using the top eight or the bottom eight of the FX-80's nine pins. See your FX-80 manual for details. Following each attribute byte are the 11 data values for the character.

Because only seven-bit values can be sent from an Applesoft program, the character was defined using the bottom seven pins. If you want to send eight-bit values, you can use a hybrid program similar to the one used for the ImageWriter (Listings 3 and 4).

ImageWriter

The process for designing a custom character for the ImageWriter is similar to the process for the Epson FX-80. In this case, the carriage return symbol replaces the ampersand (&), but only temporarily. The switch is made, the symbol printed, and the standard definition is reinstated. The rest of the time you have full access to all standard characters, including the ampersand. The ImageWriter commands are entirely different from Epson commands and the pins are numbered in the opposite direction. Figure 3 shows how an ImageWriter character is designed and coded. Unlike the Applesoft program used for the FX-80 character (above), the machine language portion of IMAGEWRITER.PC (Listing 4) can send ASCII characters 128 and higher. If you need to send data values greater than 127, be sure to set the DIP switch that allows your ImageWriter to recognize the eighth bit. This example requires only seven bits.

GENERATING LINE NUMBERS

Since WPL programs cannot exceed 2,048 characters in length, they are seldom longer than 999 lines. This is significant because WPL allows for only three numerical variables (X, Y and Z), which is sufficient to simulate a three-digit decimal counter.

The simulation involves assigning X, Y and Z to the units, tens, and hundreds positions, respectively. Each variable is se-

quenced from 0 to 9, in steps of one, as many times as required. The counter behaves like the odometer in your car — each "revolution" of X causes Y to advance by one.

The line numbers produced by the counter, and a colon, are inserted after each carriage return in the WPL file (lines 32-33). To accommodate the first line, a carriage return is temporarily inserted at the beginning of the file (lines 77-79). The last line requires special treatment because it may or may not end with a carriage return. A colon is temporarily inserted at the end of the file to handle this situation (lines 12-14).

HOW THE PROGRAM WORKS

Lines 8-10 turn off the text editor display, assign an empty string to the variable SC, and jump to ST. The NR subroutine (lines 11-45) generates line numbers for the file that is being processed. This subroutine is located near the beginning of the program to speed execution.

Lines 11-17 set the left margin to eight, insert a colon at the end of the file, and print the message "Generating line numbers." Lines 18-31 employ the X, Y and Z variables to simulate a three-digit decimal counter. Then lines 32-34 insert a line number and colon after each carriage return in the file and return to LP (line 21) if there are any carriage returns remaining.

Lines 35-37 remove the carriage return that had been temporarily inserted at the beginning of the file. Lines 38-45 remove any extraneous characters from the end of the file and return to the line following the call to the subroutine.

The program actually begins at line 46, which clears the WPL display. Line 47 prints the name of the program at the top of the screen. After skipping a line (line 48), line 49 prompts for the name of the file, which is stored in variable SA. Lines 50-

53 check the input for {esc} (which terminates the program) and for an empty string (an invalid file name). If the input is invalid, the prompt is repeated on the next line of the display.

Line 55 prompts for the drive number, which is stored in variable SB, and lines 56-60 check the input for a valid drive number. Lines 62-64 check the variable SC for a previously entered date. If it contains an empty string, a date has not yet been entered and the program jumps to D2; otherwise, it jumps to OP.

Line 65 prompts for the date, which is stored in variable SC. Lines 66-67 check the input for an empty string, which is not a valid date. Note the use of the asterisk (*) as a delimiter in line 66, since the slash (/) is frequently used for specifying the date.

Line 69 prompts for a choice of option, which is stored in variable SD, and lines 70-74 check the input for a valid selection.

Lines 75-79 clear memory, load the file, and insert a carriage return at the beginning of the file. Lines 80-93 establish all of the page parameters for printing, except for the left margin, which is handled later on. Note the inclusion of the file name (SA) and date (SC) in the specification for the header in line 92.

In line 94, the variable SB is assigned the character string "replaced" (with spaces before and after the word). This will be part of the message that is displayed each time a character substitution is made.

Lines 95-96 clear the WPL display and print the message "Substituting characters." Then lines 98-117 replace {esc}, [A], [G], [K], [L], [N], [O], [Q], [R] and [T] with the appropriate character strings, everywhere in the file. Each string begins with

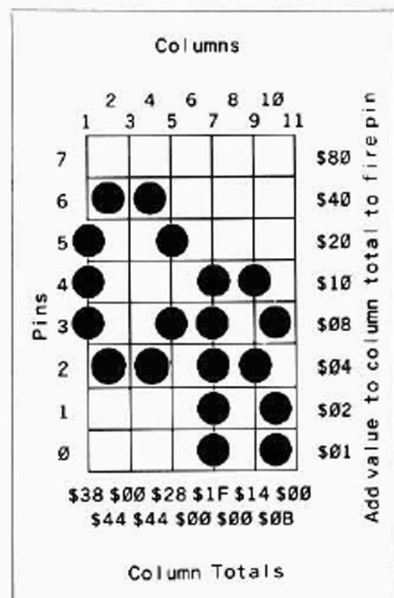


FIGURE 2: Programmed Character for Epson FX-80

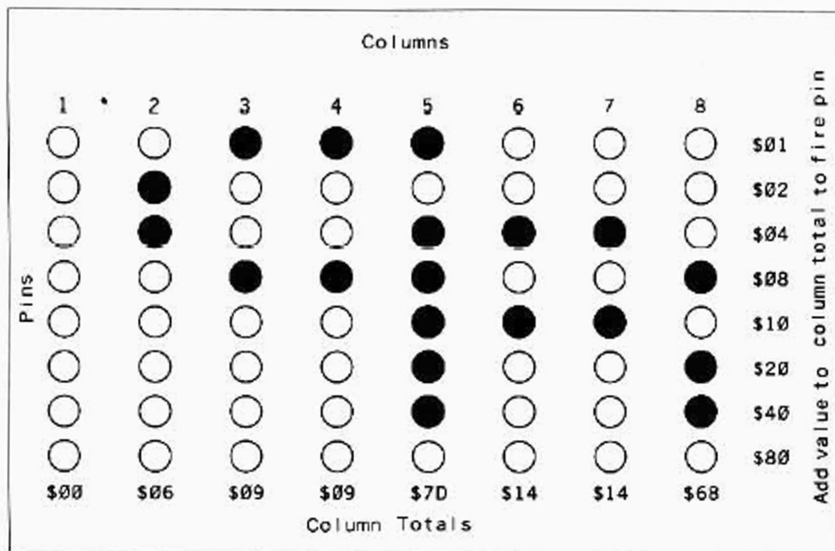


FIGURE 3: Programmed Character for Apple ImageWriter

{esc}E (to turn on emphasized print) and ends with {esc}F (to turn off emphasized print). Only when a substitution is made is a message actually displayed on the screen (e.g., line 99 displays "{esc} replaced" only when the file contains at least one {esc} character).

Lines 118-123 handle the footnotes and backslash characters. Since footnotes are delimited by pairs of characters, they are merely separated from each other by the [Q] to facilitate printing.

Lines 124-127 check the variable \$D to see if option 1 (All) has been selected. If so,

the program jumps to AL; otherwise, it calls the subroutine NR (line 11) to generate line numbers and then jumps to PR (line 127).

WPL files frequently contain control characters and other special characters that interfere with normal printing.

Lines 128-129 set the left margin to five and position the text editor cursor at the beginning of the file. Lines 130-131 insert a control Q ([Q]) before all embedded commands, which are identified by a carriage return immediately followed by a period. This translates to "%." being replaced by "%[Q]." in line 128.

Lines 132-133 remove the carriage return that was previously inserted at the beginning of the file. Then lines 134-135 replace all of the remaining carriage returns with {cr}. Lines 136-138 insert a form feed (.ff) command at the end of the file.

Line 136 initiates the printing process. When this is done, line 141 loops back to ST to prompt you for a new file name. Finally, line 143 clears memory before exiting the program.

THIS PROGRAM IS AVAILABLE ON DISK

If you'd rather not type in the listing for this program, you can buy it on disk, complete, free of typos and ready to run. The Reviewer, DISASTEP, ProDOS Position Command, Applesoft Band-aids and Remote Control Apple Writer are available on disk for an introductory price of \$17.95 plus \$1.50 shipping/handling (\$2.50 outside the U.S.) from Nibble, 45 Winthrop St., Concord, MA 01742. Introductory price expires 3/31/87. See the coupon on the last page of the Nibble Software Catalog for ordering information.

LISTING 1: WPL.PRINT for Epson Printers and DOS 3.3 Apple Writer

```

001: P .....
002: P * WPL.PRINT *
003: P * By I. Larry Morris *
004: P * Copyright (C) 1987 *
005: P * By MicroSPARC, Inc *
006: P * Concord, MA 01742 *
007: P .....
008: PND
009: PAS=$C
010: PGO ST
011: NR PLM 8
012: E
013: F///
014: Y?
015: B
016: PPR
017: PPR Generating line numbers ...
018: PSX 0
019: PSY 0
020: PSZ 0
021: LP PSX +1
022: PCS /10/(X)/
023: PGO IY
024: PGO AP
025: IY PSX 0
026: IY PSY +1
027: PCS /10/(Y)/
028: PGO IZ
029: PGO AP
030: IZ PSY 0
031: PSZ +1

```

```

032: AP F<><>(Z)(Y)(X): <
033: Y?
034: PGO LP
035: B
036: F<><<
037: Y?
038: E
039: F<??:??: :<<
040: Y?
041: PRT
042: E
043: F///
044: Y?
045: PRT
046: ST PPR [L]
047: PPR
048: PPR
049: FL PIN Enter file name (ESCAPE = Quit): =$A
050: PCS /$A/{esc}/
051: PGO QT
052: PCS /$A//
053: PGO FL
054: PPR
055: DR PIN Enter drive no. (1 or 2): =$B
056: PCS /$B/1/
057: PGO DT
058: PCS /$B/2/
059: PGO DT
060: PGO DR
061: DT PPR
062: PCS -$C++
063: PGO D2
064: PGO OP
065: D2 PIN Enter date: =$C
066: PCS -$C++
067: PGO D2
068: PPR
069: OP PIN Select option (1 = All, 2 = WPL): =$D
070: PCS /$D/1/
071: PGO LD
072: PCS /$D/2/
073: PGO LD
074: PGO OP
075: LD NY
076: L $A,DSB
077: B
078: F<><<
079: Y?

```

LISTING 1: WPL.PRINT for Epson Printers and
DOS 3.3 Apple Writer (continued)

```

080: PPM 0
081: PRM 80
082: PTM 1
083: PBM 0
084: PPN 1
085: PPL 60
086: PPI 66
087: PLI 0
088: PSP 0
089: PPD 1
090: PCR 0
091: PLJ
092: PTL-+{ESC}E$A ... 3C{ESC}F-Page #+
093: PBL
094: PAS replaced =SB
095: PPR [L]
096: PPR Substituting characters ...
097: PPR
098: F/{ESC}/{ESC}E{ESC}{ESC}F/A
099: PPR {ESC}SB
100: F/[A]/[ESC]E[A]{ESC}F/A
101: PPR ctrl-ASB
102: F/[G]/[ESC]E[G]{ESC}F/A
103: PPR ctrl-GSB
104: F/[K]/[ESC]E[K]{ESC}F/A
105: PPR ctrl-KSB
106: F/[L]/[ESC]E[L]{ESC}F/A
107: PPR ctrl-LSB
108: F/[N]/[ESC]E[N]{ESC}F/A
109: PPR ctrl-N$B
110: F/[O]/[ESC]E[O]{ESC}F/A
111: PPR ctrl-OSB
112: F/[Q]/[ESC]E[Q]{ESC}F/A
113: PPR ctrl-QSB
114: F/[R]/[ESC]E[R]{ESC}F/A
115: PPR ctrl-R$B
116: F/[T]/[ESC]E[T]{ESC}F/A
117: PPR ctrl-T$B
118: F/(<[Q]/<A
119: PPR (<SB
120: F/> >[Q]/A
121: PPR > >SB
122: F/[backslash]/[ESC]E[backslash]{ESC}F/A
123: PPR [backslash]SB
124: PCS /SD/1/
125: PGO AL
126: PSR NR
127: PGO PR
128: AL PLM 5
129: B
130: F#.#%[Q].#A
131: PPR $B
132: F<<<<
133: Y?
134: F<<<[ESC]E[cr][ESC]F>>>A
135: PPR [cr]SB
136: PR E
137: F<<<.ff<
138: Y?
139: PPR
140: PNP
141: PGO ST
142: QT PPR
143: NY
144: PIN End of PRINT (Press RETURN)
END OF LISTING 1

```

LISTING 2: FX80.PC

```

10 REM *****
20 REM = FX80.PC *
30 REM = BY LOREN W. WRIGHT *
40 REM = COPYRIGHT (C) 1987 *
50 REM = BY MICROSPARC, INC *
60 REM = CONCORD, MA 01742 *
70 REM *****
80 REM RUN THIS PROGRAM
90 REM BEFORE BOOTING APPLEWRITER
100 E$ = CHR$(27); D$ = CHR$(4); C0$ = CHR$(0); C1$ = CHR$(1); CQ$ = CHR$(17)
110 PRINT D$"PR#1": PRINT CHR$(9)"80N"
120 PRINT E$"C1$C0$:: REM SELECT RAM CHARACTER SET
130 PRINT E$"C0$C0$C0$:: REM COPY ROM CHARACTERS TO RAM
140 PRINT E$"I1": REM ENABLE PRINTING OF CHARACTERS 0-31
150 PRINT E$"C0$C0$C0$ CHR$(26):: REM START CHARACTER DEFINITION
160 FOR I = 1 TO 11: READ N: PRINT CHR$(N):: NEXT
170 PRINT : PRINT D$"PR#0"

```

```

180 HOME : VTAB 12: PRINT "FX-80 <CTRL>Q REPLACED": PRINT "PRESS SPACE TO TEST. <RETURN> TO QUIT": GET Z$: PRINT : IF Z$ < > " " GOTO 200
190 PRINT D$"PR#1": PRINT CQ$CQ$CQ$
200 PRINT D$"PR#0"
210 DATA 56,68,0,68,40,0,31,0,20,11,0
END OF LISTING 2

```

LISTING 3: IMAGEWRITER.PC

```

10 REM *****
20 REM = IMAGEWRITER.PC *
30 REM = BY JAN EUGENIDES *
40 REM = COPYRIGHT (C) 1987 *
50 REM = BY MICROSPARC, INC *
60 REM = CONCORD, MA 01742 *
70 REM *****
80 REM RUN THIS BEFORE BOOTING APPLEWRITER
90 TEXT : NORMAL
100 S1$ = "300:A2 00 BD 11 03 C9 FF F0 07 20 ED FD E8 4C 02 03 60 1B 2D 1B 49 A6 48 0 0 06 09 16 09 09 16 09 7D 14 14 68 04 FF N D823G"
110 HOME : HTAB 8: PRINT "CUSTOM CHARACTER INSTALLER": HTAB 3: PRINT "COPYRIGHT 1986 BY MICROSPARC, INC."
120 VTAB 12: HTAB 8: PRINT "TURN ON YOUR IMAGEWRITER": VTAB 22: PRINT "PRESS <RETURN> WHEN READY ": GET CH$: PRINT CH$
130 HOME : VTAB 10: PRINT "UPLOADING CHARACTER TO IMAGEWRITER..."
140 FOR X = 1 TO LEN(S1$): POKE 511 + X, ASC(MID$(S1$,X)) + 128: NEXT
150 POKE 72,0: CALL 65392
160 PRINT CHR$(4)"PR#1"
170 CALL 768: PRINT
180 PRINT CHR$(4)"PR#0"
190 HOME : VTAB 9: HTAB 10: PRINT "CHARACTER INSTALLED"
200 PRINT : PRINT "PRESS <RETURN> TO TEST, <ESC> TO END"
210 GET CH$: PRINT CH$: IF CH$ = CHR$(27) THEN GOTO 250
220 PRINT CHR$(4)"PR#1"
230 PRINT CHR$(27): CHR$(42): "&": CHR$(27): CHR$(36)
240 PRINT CHR$(4)"PR#0"
250 TEXT : END
END OF LISTING 3

```

LISTING 4: IMAGEWRITER.PC Assembly Code

```

1000 *****
1010 * IWRITER.PC.ML
1020 * A SIMPLE LOOP TO UPLOAD HEX
1030 * CODES TO IMAGEWRITER PRINTER
1040 * THIS ROUTINE IS CALLED BY
1050 * THE CHAR UPLOADER PROGRAM IN
1060 * LISTING 3
1070 *
1080 * BY JAN EUGENIDES
1090 * COPYRIGHT 1987
1100 * BY MICROSPARC, INC.
1110 * CONCORD, MA 01742
1120 * S-C MACRO ASSEMBLER
1130 *****
1140 .OR $300 A FAVORITE LOCATION
1150 COUT .EQ $FDED MONITOR CHARACTER OUTPUT
1160 START
0300- A2 00 1170 LDX #0 INITIALIZE INDEX
0302- 0D 11 03 1180 .I LDA DATA,X GET DATA BYTE
0305- C9 FF 1190 CMP #4FF IS IT $FF?
0307- F0 07 1200 BEQ END YES, ALL DONE
0309- 20 ED FD 1210 JSR COUT NO, SEND IT OUT
030C- E8 1220 INX INCREMENT INDEX
030D- 4C 02 03 1230 JMP .I BACK TO TOP OF LOOP
0310- 60 1240 END RTS RETURN TO CALLER
0311- 1B 2D 1B
0314- 49 1250 DATA .HS 1B.2D.1B.49 CONTROL CODES FOR IWRITER
0315- A6 1260 .AS "-"&" & WITH HIGH BIT SET
0316- 48 00 06
0319- 09 16 1270 .HS 48.00.06.09.16 MUST REPLACE ^I W/ ^V
031B- 09 09 16
031E- 09 7D 1280 .HS 09.09.16.09.7D TO GET 09 DATA OUT
0320- 14 14 68
0323- 04 FF 1290 .HS 14.14.68.04.FF ENDS WITH ^D
END OF LISTING 4

```