



Apertext Extended Text System User Manual

(C) Copyright Apex Software 1984

Table of Contents

INTRODUCTORY WORDS AND LEGAL STUFF	1
Why Use Apertext?	1
Using Apertext in a BASIC program	2
THE GENERATOR (APEXTEXT)	3
Character Parameters	3
Font Size	4
Bold	4
Inverse/Normal	5
Underlined Text	5
Block Graphics	5
Font Type	6
Character and Background Color	7
Cursor Controls	7
Screen Functions	7
Other Functions	8
Large Character Special Functions	8
Overstrike and Turning Off Scrolling or Clear	8
THE EDITORS	8
Disk Commands	10
Edit Window Commands	10
Edit System Commands	11
Swapping Characters	11
SUPPORT PROGRAMS AND ROUTINES	12
Puzzle Maker	12
Font Printer	12
Screen Machine	12
Apex Input Routine	13
Scrolling Routines	13
TECHNICAL NOTES	14
Memory Map	14
Miscellaneous Notes	14
ProDOS Apertext	14
Double Hi-Res Apertext	15
One Final Word	15

INTRODUCTORY WORDS AND LEGAL STUFF

Disclaimer: This manual and accompanying diskette are available only from Apex Software and Apex dealers. Every effort has been made to provide error-free programs and documentation. Inevitably, some may remain. Apex Software denies any responsibility for loss or damage of programs or equipment, direct or indirect, even if Apex Software has been advised of such damages.

Notice: Apex Software reserves the right to make improvements in the product described in this manual at any time and without notice. The word Apple is a registered trademark of Apple Computer, Inc. Apple Computer, Inc. makes no warranties, either expressed or implied, regarding the enclosed software package, its merchantability or its fitness for any particular purpose. DOS 3.3 is a copyrighted program of Apple Computer, Inc. licensed to Apex Software for use only in combination with Apertext. Apple Software shall not be copied onto another diskette (except for archive purposes) or into memory unless as part of the execution of Apertext. When Apertext has completed execution Apple Software shall not be used by any other program.

This manual gives a simple, logical explanation of the use of Apertext (EXtended TEXT system) and its support programs. It assumes the user has at least a knowledge of Applesoft BASIC programming and simple DOS commands. Apertext comes unprotected and LISTable so that you may use it in your own programs and back it up easily. Make a back-up copy immediately so as to protect your investment, but please don't make copies and distribute it to the whole world. That's our job at Apex Software. Apertext is a copyrighted program, but it is possible to use Apertext in your own programs for profit. To obtain a license to do this contact Apex Software directly.

Why Use Apertext?

Apertext is a very fast and versatile character generator. It generates characters, letters, symbols, or pictures on the video screen. You may print characters in many styles including: normal, wide, tall, expanded, bold, inverse, colored, and underlined. Apertext also provides routines for sound, clearing the screen and scrolling in four directions. It is compatible with ProDOS, DOS 3.3, Applesoft, machine language, double hi-res, all Apple // series computers, and fonts from many other software packages.

Most people that use Apertext will be programmers. Apertext works to save the programmer development time. Three example applications of Apertext are included on the system disk. Studying how these programs work may give more insight into the use of Apertext than any manual could provide no matter how well written. The programs are called "DEMO.APEX", "MAGIC" and "MAGIC1". Try LISTing them and RUNning them.

Using Apertext in a BASIC program

Once Apertext is loaded in memory, it functions using normal Applesoft commands. The following program will load Apertext and get it started. The program may be found on the Apertext system disk named "APEX.SETUP".

```
100 LOMEM: 30464: AP=24576: SM=26624: LG=27392
110 D$=CHR$(4): PRINT D$"BLOAD APEX.32"
120 PRINT D$"BLOAD ASCII.SF,A"SM
130 PRINT D$"BLOAD ASCII.LF.LC,A"LG
140 B$=CHR$(2): C$=CHR$(3): EX$=CHR$(5): S$=CHR$(6): BL$=CHR$(7):
```

```

    IN$=CHR$(9): HM$=CHR$(12): CL$=CHR$(14): CS$=CHR$(15):
    HB$=CHR$(16): LG$=CHR$(17): RG$=CHR$(18)
150 SF$=CHR$(19): TL$=CHR$(20): CK$=CHR$(22): WD$=CHR$(23):
    B1$=CHR$(25): B2$=CHR$(26): US$=CHR$(28): BD$=CHR$(29):
    UD$=CHR$(30): CD$=CHR$(31)
160 CALL AP: POKE AP+33,(SM/256)-1: POKE AP+34,(LG/256)-4
170 REM

180 FOR A=32 TO 127: PRINT CHR$(A);: NEXT
190 PRINT LG$: FOR A=32 TO 127: PRINT CHR$(A);: NEXT: PRINT:
    PRINT RG$

```

RUN the program with the Apertext disk in the drive. It should PRINT the small and large character set on the screen and then leave you in Applesoft BASIC. The output should look similar to that shown below:

```

! "#$%&'()*+,-./0123456789:;<=>?@ABCDEFGH
IJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnop
qrstuvwxyz{|}~

```

```

! "#$%&'()*+,-./0123
456789:;<=>?@ABCDEFGH
IJKLMNOPQRSTUVWXYZ[
\]^_`abcdefghijklmnop
qrstuvwxyz{|}~

```

]

This exercise demonstrates a program that starts Apertext and does some very simple commands to PRINT characters. A line by line discussion of the program is in order here. Line 100 sets the lowest memory location that will be available for Applesoft variables called LOMEM. Line 100 sets the memory locations for Apertext, a small font, and a large font. Line 110-130 loads the necessary files from the disk. Line 140-150 set up standard string variables used for Apertext functions. Line 160 initiates Apertext by turning on the graphics screen and setting up font locations in memory. Line 180 PRINTs the small font. Line 190 concludes the program by changing to the large font, PRINTing that font, and changing back to the regular font size. Perhaps the most difficult steps are in line 100 and 160 where you must understand a good deal about memory mapping on the Apple. This can be avoided by simply using the setup program "as is". (The only time memory mapping becomes a concern is in extremely large programs or when machine language subroutines conflict with Apertext.) Just begin writing your application by removing lines 180 and 190 and add your own program lines from there on. If you wish to disable Apertext at any time you may press RESET or add line 200 below to your program. Line 210 directly below line 200 will warmstart Apertext without resetting its default values and clearing the screen.

```

200 CALL AP+6: TEXT
210 CALL AP+3

```

THE GENERATOR (APEXTEXT)

The generator is the program named "APEX.32" on the Apertext diskette. This program as well as a character font may be copied to another disk using a file utility program such as COPY II+. A font is a specific Apertext character set. Different fonts may give characters or letters a different look or completely redefine the characters to be graphic pictures. The two standard fonts are: "ASCII.SF" (normal small font) and "ASCII.LF.LC" (large font with lower case). Functions within Apertext are chosen by PRINTing the proper control character in a BASIC program, by typing the control character directly from the keyboard, or through COUT in assembly language. Control characters in BASIC programs may be printed in three ways:

1. Entering the character directly within the quotes of a PRINT statement.
2. PRINTing the CHR\$(n) equivalent of the control character.
3. Assigning the control character to a variable and then PRINTing that variable.

Method number three is preferred because it allows you to assign variables that give meaning to the control characters, the variable shows-up on the screen for easy editing, and the variable will not change printing modes when you list the program on the screen or on a printer (whereas the direct entry method may).

Character Parameters

There are many character parameters to choose from. These are presented in outline form and then explained in more detail for simplicity:

- A. Font Size (Large or Small)
- B. Display Mode
 1. Large Font Only -- Shifted output
 2. Small Font Only
 - a. Double Wide
 - b. Double Height
 - c. Expanded (Double Width and Height)
 3. Both Large and Small Font
 - a. Bold/Normal
 - b. Inverse/Normal
 - c. Underlined
 - d. Block graphics
- C. Font Type
- D. Font Color
- E. Background Color
- F. Upper/Lower Case
- G. Overstrike

Font Size

The following control characters change the font size:

CHR\$(5)	CTRL-E	Select small font Expanded mode
CHR\$(17)	CTRL-Q	Select large font
CHR\$(18)	CTRL-R	Select small font Regular mode
CHR\$(20)	CTRL-T	Select small font Tall mode
CHR\$(23)	CTRL-W	Select small font Wide mode



Font and character sizes available (from APEX.DEMO program screen).

Bold

Choosing bold type produces a character that is thicker than normal. For example: **BOLD NOT BOLD**. The generator's bold mode may be used by printing the following control characters:

CHR\$(25) CTRL-Y Bold on
 CHR\$(26) CTRL-Z Bold off

Inverse/Normal

When the INVERSE mode is in effect, all characters output will be reversed. For example: a character's specified color is blue and its background is white. In INVERSE this character will be white and have a blue background. The following text demonstrates:

INVERSE **NORMAL**

Although the Applesoft commands INVERSE and NORMAL are operational under Apextext, it is sometimes easier to use:

CHR\$(9) CTRL-I Toggle inverse/normal

Every time this character is PRINTed it toggles to the opposite of the current inverse/normal status.

Underlined Text

Apextext provides an underline display mode. There are some minor limitations on underlining. First, underlined words must be double spaced because underlined characters

may not have text PRINTed directly below them or the underlines will be wiped out. Secondly, underlined characters may not be PRINTed on the bottom line of the screen.

CHR\$(30) CTRL-^ Turn underlining on/off
CHR\$(31) Turn underlining off ignoring the previous setting

See the following programming example:

```
500 PRINT "No Underlines "CHR$(30)"Underlines"CHR$(31)
```

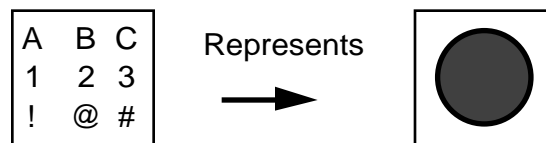
Will appear on the screen as: No Underlines Underlines

Block Graphics

Perhaps the most fun and useful output mode, block graphics allows you to perform fast, high resolution graphics from BASIC or machine language. The main concept behind block graphics is that more than one character may be used to define a hi-res picture and this “block” of characters may be displayed without separate HTABing and VTABing for each character of the block. The commands for this are summarized below:

CHR\$(29) CTRL-] Turn on block display mode
CHR\$(31) Turn off display modes or end block definition
CHR\$(16) CTRL-P Home block cursor

PRINTing a CHR\$(29) will start a block definition by saving the current HTAB and VTAB locations within Apextext. Following this any character may be PRINTed (the first line of block definition) until Apextext recognizes a CHR\$(13) which causes a carriage return to the saved HTAB position. In other words, the next character to be PRINTed will be directly below the first character PRINTed (“A” in this case). This may then be repeated. To end the block definition PRINT CHR\$(31). To restore the cursor to the saved values simply PRINT CHR\$(16) to home the block cursor. (This home will not clear the screen but rather resets the block cursor position.) For example, you use the characters “ABC”, “123”, and “!@#” as the first, second and third line of the picture of a circle. The circle is created using the a character editor to make parts of the circle picture in the characters “ABC123!@#”.



This circle may be PRINTed on the screen using block graphics mode by the following line:

```
1000 PRINT CHR$(29) "ABC"CHR$(13)"123"CHR$(13)"!@#"CHR$(31)
```

As an exercise, we may replace the circle picture by three words. Home the block cursor and PRINT a block of nine characters by:

```
1010 PRINT CHR$(16) "CAT"CHR$(13)"THE"CHR$(13)"DOG"CHR$(31)
```

Font Type

The font type or style of lettering is chosen by loading different fonts from the disk into a safe location in memory and then POKEing the correct font index. The Apertext disk contains at least 16 fonts (See Font styles at end.) The formula for calculating a font index is given in the "APEX.SETUP" program. The user simply sets SM (for a small font) or LG (for a large font) equal to the starting memory location of the font and then lets line 160 do all the work. There is a table of five small/large font combinations within Apertext. By POKEing into the table and PRINTing CTRL-S or CHR\$(19) followed by a number 0-4 these fonts may be switched back and forth at will within a PRINT statement. The 4 other font combination locations are directly after AP+33 and AP+34 (locations AP+35 through AP+42) and are located as pairs of one small font index followed by one large font index.

The default locations for a font used by Apertext are \$1D00 for small fonts and \$1400 for Large Fonts. All saved fonts load at these locations unless specified. Apertext has many fonts included on the system disk. In addition, you may use fonts from many other commercial character generators. Fonts may be transferred to another disk with a disk utility or by BLOADing the file and then BSAVEing it on the other disk. The following commands will save an Apertext font after you have BLOADed it:

```
BSAVE [font name],A$1400,L$C00      (for a large font w/lower case)
BSAVE [font name],A$1400,L$800      (for a large font w/o lower case)
BSAVE [font name],A$1D00,L$300      (for a small font)
```

Character and Background Color

The character color is selected by the prefix CHR\$(3) or CTRL-C followed by a number (0-7) for the standard apple colors or by a letter (A-X) for additional color shades and mixes. The background may be set to the current color by PRINTing a CHR\$(2) or CTRL-B. Immediately after the background color has been set, the new character color should be set by PRINTing CHR\$(3) and the color code (0-7 or A-X). Using the 32 different colors available to the user and the same 32 colors as different backgrounds there are 2048 different possible text color/background combinations, which is more than enough to choose from for any application. (Note: Using certain color/background combinations may not produce the exact color that you expected due to limitations of the Apple video display.)

```
CHR$(2)   CTRL-B   Set background to current color
CHR$(3)   CTRL-C   Set color (follow by number 0-7 or letter A-X)
```

Cursor Controls

Apertext has four control characters reserved exclusively for cursor control. These are:

```
CHR$(8)   CTRL-H   Backspace (Left)
CHR$(10)  CTRL-J   Linefeed (Down)
CHR$(11)  CTRL-K   Reverse Linefeed (Up)
CHR$(21)  CTRL-U   Forward Space (Right)
```

Screen Functions

Apextext provides 4 built-in screen functions:

CHR\$(12)	CTRL-L	Clears screen, home cursor (HOME)
CHR\$(14)	CTRL-N	Clears current line from cursor to end of line
CHR\$(15)	CTRL-O	Clears from current cursor position to end of page
CHR\$(6)	CTRL-F	Scrolls text up one line

The normal screen function CALLS and HOME command for clearing the screen will not work with Apextext. They will succeed in simply clearing the TEXT screen and not the hi-res screen. For example: Instead of using the HOME command, PRINT CHR\$(12), which performs the same function. These routines also take the background color into consideration and clear the proper window area to the current background color. The window area is selected normally by these POKES which are standard for the Apple:

```
POKE 32, Window Left-Edge (0-39)
POKE 33, Window Width (1-40)
POKE 34, Window Top-Edge (0-23)
POKE 35, Window Bottom (1-24)
```

Other Functions

There are other miscellaneous functions that Apextext performs as shown below:

CHR\$(4)	CTRL-D	Reserved for DOS use
CHR\$(7)	CTRL-G	Modified bell tone
CHR\$(13)	CTRL-M	Carriage Return
CHR\$(22)	CTRL-V	Click speaker
CHR\$(28)		User Function

Lower case is selected by printing the correct character (ASCII value 97-122 or 225-250). Lower case may be typed directly into your program on the Apple IIe/IIc/IIgs. If you own an Apple II+ you may POKE AP+66,128 (AP = starting address of Apextext) which allows you to enter lower case by using the CTRL-A as a toggle for upper/lower case. Another method of entering lower case is by using a program line editor. The bell tone and click functions double as powerful sound routines. Three POKES change the sounds of CTRL-G and CTRL-V. These are shown on the quick reference chart.

Large Character Special Functions

The large characters may be shifted one bit left before being printed on the screen. This has the effect of changing green to violet, violet to green, orange to blue, and blue to orange. You may do this by POKEing AP+64 to 0 for normal or 128 for shifted large characters. The hi-bit may be set by using color number 7, and cleared by using color number 3. Setting the hi-bit changes green/violet to orange/blue.

Overstrike and Turning Off Scrolling or Clear

Overstrike mode allows characters to be PRINTed over eachother or over backgrounds. It is invoked by a POKE AP+69,128 and turned off by POKE AP+69,0. It is useful also to create special characters. For example, PRINTing “/”CHR\$(8)“\” will result in an “X” because the slashes will overstrike eachother. Other Apextext functions which may be useful are: POKE AP+70,128 which turns scrolling off, and POKE AP+65,128 which

turns clear to end of line (CEOL) off after a carriage return.

THE EDITORS

Apertext has two character editors, one for small fonts (EDIT.SM) and one for large fonts (EDIT.LG). Both editors work essentially the same. The differences are outlined below:

1. EDIT.LG will only edit fonts with the suffix “.LF” or “.LF.LC” correctly
2. EDIT.SM will only edit fonts with the suffix “.SF” correctly
3. EDIT.SM has provisions for 2 character sets in the memory at the same time



EDIT.LG screen.



EDIT.SM screen.

The basic concept behind editing a font is to change the style or shape of one or more characters in the font or to create a new font altogether. This is done by:

1. Loading the font to be worked on from the disk ([L] loads)
2. Pulling the character to be edited into the edit window ([I], [J], [K], [M] select character and [P] pulls into window)
3. Editing or changing the character (many keys)
4. Returning the character from the edit window back into the font ([I], [J], [K], [M] select character and [RETURN] puts it back)
5. Saving the font back out to the disk ([O] outputs)

Disk Commands

The following one letter commands are supported from the editors for saving and loading fonts:

- | | |
|---|----------------------------|
| D | Disk CATALOG |
| L | LOAD Font from disk |
| O | Output (SAVE) Font to disk |

When loading or saving files, the full name of the file must be used including the suffix. The suffix “.SF” stands for Small Font, “.LF” for Large Font, and “.LC” for Lower Case. Since large fonts take up four times as much memory as a small font, they may be saved with or without lower case. If saved without the lower case enough memory is saved that an entire small font could occupy the unused space. For example, the font “ASCII.LF.LC” is a large font with lower case and occupies more memory than the “CARD.LF” font which is saved without lower case. “ASCII.SF” is a small font and

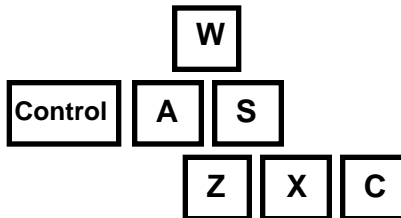
must contain lower case.

Edit Window Commands

The edit window is the box at the top of the screen that resembles a graph. Edit window commands will only affect what is in the edit window. The edit window cursor is the flashing dot or box. These one letter commands control operations within the edit window:

W	Edit cursor up
A	Edit cursor left
S	Edit cursor right
Z	Edit cursor down
X	Marks the spot under cursor
C	Clears the spot under cursor
CTRL-W	Scroll up
CTRL-A	Scroll left
CTRL-S	Scroll right
CTRL-Z	Scroll down
CTRL-R	Reverse (Flip Horizontally)
CTRL-F	Flip upside down (Flip Vertically)
CTRL-I	Inverse
CTRL-B	Begin again (Clears Edit Window)

Notice the Keyboard layout. The [W], [A], [S], and [Z] keys are in a diamond shape indicating the direction they move, and the scroll keys are the same except Control is held down simultaneously. The [X] and [C] keys for turning dots on and off are right nearby.



Edit System Commands

These other miscellaneous commands may be keyed to produce other useful functions:

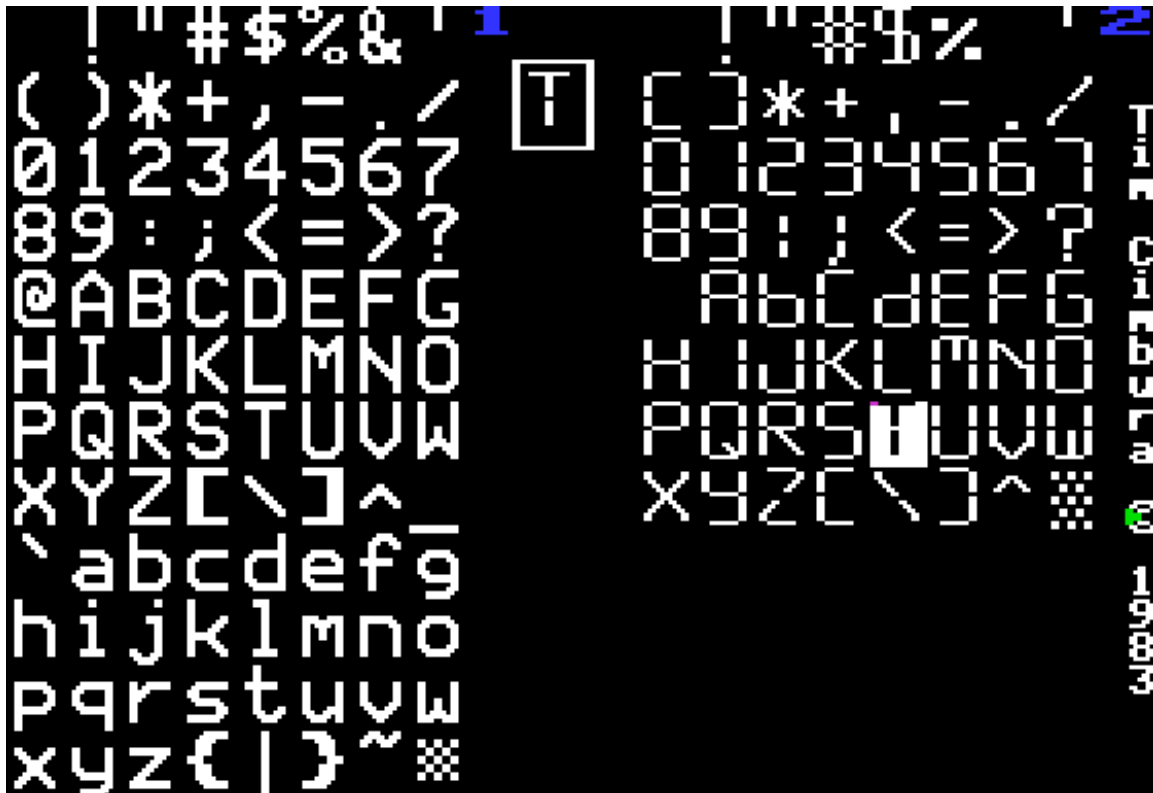
P	Pull character into edit window
[RETURN]	Save character in font
CTRL-C	End program (All fonts will still be intact)
CTRL-L	Toggle large font lower case on/off
1	Select character set #1
2	Select character set #2
I	Font Cursor Up
J	Font Cursor Left
K	Font Cursor Left
M	Font Cursor Down
?	Get help screen

The font cursor is used to select the character that you wish to edit. Pressing [P] will Pull the character into the edit window. After editing the character, it also determines where that

character will be placed. If you wish to return the character elsewhere in the font, move the font cursor to that spot before pressing [RETURN] to save the character in the edit window in the font.

Swapping Characters

Moving a character from one font to another may be accomplished very easily. In the small font editor, simply load the two fonts as [1] and [2]. Press the appropriate key for which font the character is from, then do a [P] (Pull) into the edit window. Then select the other font also with a [1] or [2] and [RETURN] the character to the font. To swap large font characters, you must use the large font swapper (SWAP.LG) and follow the same procedure.



SWAP.LG screen.

SUPPORT PROGRAMS AND ROUTINES

Puzzle Maker

PUZZLE.MAKER is a utility program on the Apertext disk that converts anything on the hi-res screen into either a large or small font. In this way a shape table font may be easily converted to an Apertext font by first drawing it on the screen and then RUNNING PUZZLE.MAKER. The program FIX.FONTS will draw a shape table font on the screen in a perfect format for PUZZLE.MAKER. Fonts from other graphics systems or character generators may operate with Apertext with little or no modification.

Font Printer

FONT.PRINTER is a program that prints signs and messages on any printer using any

font as the type style. Try this by selecting the large font WESTERN.LF.LC in 4 by 4 and printing a "WANTED" sign.

Screen Machine

SCREEN is a very useful utility for creating slides or pictures with Apertext graphics. The program allows the user to (1) Safeguard/Recover picture in memory, (2) Overstrike, (3) LOAD/SAVE hires picture, (4) Load new fonts, (5) CATALOG either drive, and (6) Type anything over or on any hires picture. Commands to change modes are entered directly from the keyboard in the sequence "normal" to Apertext. For example, to change the color type CTRL-C and then the color code (0-7 or A-X). For ease of use, a click function is included allowing each key typed and accepted to be heard.

Apex Input Routine

INPUT.APEX is a short (under 256 bytes) machine language input routine. It is relocatable, so that it may be loaded and executed anywhere in memory. To load you should set IN=(the starting address of the input routine) and BLOAD INPUT.APEX,A (that address). Thereafter an input may be performed by a CALL IN,(length of input). The length should be between 1 and 250 inclusive. The input is returned to BASIC in I\$. This routine performs flawlessly in any Apertext mode or color.

Scrolling Routines

The scroll routines for down, left and right scrolling are named SCROLL and SCROLL.DBL on the Apertext disk. Both routines are relocatable. SCROLL.DBL works in double hi-res identically to SCROLL but occupies more memory area. See the quick reference sheet for more detail on calling procedures.

TECHNICAL NOTES

Memory Map

	Monitor	

	Applesoft	
\$D000	-----	
	I/O Addresses	
\$C000	-----	
	DOS	
	-----	HIMEM: (Set by DOS)
	Basic Variables	
\$7700	-----	LOMEM:30464
	LARGE FONT	
\$6B00	-----	SM = 27392
	SMALL FONT	
\$6800	-----	LG = 26624
	Apertext Character Generator	
\$6000	-----	AP = 24576
	Hi-Res Page 2	
\$4000	-----	16384
	Hi-Res Page 1	
\$2000	-----	8192
	BASIC Program	
\$0800	-----	2048
	Text Page	
\$0400	-----	1024
	Page 3 DOS Vectors	
\$0300	-----	768
	Input Buffer	
\$0200	-----	
	Stack	
\$0100	-----	
	Zero Page	
\$0000	-----	

Miscellaneous Notes

The hi-res page flag POKE 230,n is "smart". Apertext defaults to the page stored at 230 or chooses hi-res page 1 (\$2000-\$4000) if not previously defined. Apertext will PRINT characters of any ASCII value correctly including inverse and flashing screen codes. Apertext will not allow a program to GET a forward arrow. Apertext fools BASIC into thinking the right arrow was pressed by returning to BASIC the actual character that was retyped with the forward arrow. Therefore, an IF A\$=CHR\$(149) THEN...will not work after a GET A\$. If you need to check for the forward arrow PEEK the keyboard location.

ProDOS Apertext

The versions of Apertext that will operate with ProDOS are named APEX.PRO and APEX.PRO.DBL. These programs operate identically to their DOS 3.3 counterparts with the exception that CTRL-A toggle lower/upper case was removed. To create a ProDOS disk containing Apertext use the FILER utility to format a volume with prefix (/APEX). Next, copy the file (/USER.DISK/PRODOS) to (/APEX/PRODOS) and

(/USER.DISK/BASIC.SYSTEM) to (/APEX/BASIC.SYSTEM). Now use CONVERT to copy all files except APEX.32 and APEX.DBL from the DOS 3.3 disk to the newly created ProDOS volume with set prefix (/APEX). After all conversions are complete, exit the utility and RENAME APEX.32.PRO,APEX.32 and RENAME APEX.DBL.PRO,APEX.DBL and your disk is set for ProDOS Apertext.

Double Hi-Res Apertext

The files named APEX.DBL and APEX.DBL.PRO are the Apertext double hi-res versions. These programs operate identically to APEX.32 with a few minor exceptions. First, the program is somewhat slower since twice as much memory must be worked with to produce the graphics. Secondly, due possibly to limitations of the Apple //c, APEX.DBL has erratic operation in its input routine and clear routines. (Due to the cursor tabbing of 1-80.) This problem may be eliminated in future updates of APEX.DBL. The big advantage is in the color and background selection. Double hi-res has 16 true hi-res colors that may be mixed freely with good results. Any program operating under normal Apertext should work with APEX.DBL but with different colors shown on the screen. The color selection chart is shown below.

<i>Number</i>	<i>Code</i>	<i>Name</i>
0	0	Black
1	1	Magenta
2	2	Dark Blue
3	3	Purple
4	4	Dark Green
5	5	Grey 1
6	6	Medium Blue
7	7	Light Blue
8	A	Brown
9	B	Orange
10	C	Grey 2
11	D	Pink
12	E	Light Green
13	F	Yellow
14	G	Aquamarine
15	H	White

One Final Word

We at Apex Software wish you the best of luck with our product Apertext.