CTRL-K Caps Lock *

CTRL-L Lower Case Display

CTRL-S Shift

Normally, the Character Generator prints uppercase alphabetics in uppercase and lowercase alphabetics in lowercase. However, when the Lower Case Display mode is used, uppercase alphabetics will be printed in lowercase. This provides a convenient way for a program to print lowercase text without having to convert the data. CTRL-S prints the next character in uppercase, then puts the Character Generator in Lower Case Display mode.

CTRL-B Begin Block Display

CTRL-D Delimit Block Display

Entering Block Display mode causes the Character Generator to save the current cursor position and parameter settings. When a Block Display Delimiter (CTRL-D) is found, the original parameter settings are restored. Block Display mode is normally used in conjunction with an alternate character set to display pictures composed of a block of characters. For example, a character set might be designed to represent playing cards, with each card composed of a 2 X 2 character block. The ace of spaces might be represented by the following block of letters, each standing for one quarter of the image:

```
AS
DF
```

Although this card could be printed with two PRINT statements and appropriate VTABs and HTABs, Block Display mode could be used to represent it with the string

CTRL-B CTRL-A 1 A S CTRL-C D F CTRL-D

(The spaces are inserted for clarity, and are not part of the string.) A whole deck of such strings could be saved in an array CARD$ for use in the rest of the program.

CTRL-Z Set Default Parameters

CTRL-Z resets all of the Character Generator's parameters to their default values.

CTRL-O CTRL-Y Call User's Subroutine A

CTRL-O CTRL-Z Call User's Subroutine B

These control sequences cause the Character Generator to call a user-supplied assembly subroutine by performing a JSR. Before calling a user subroutine, the user must POKE the subroutine's address (low byte first) into the character generator at locations 1ØR and 11R ($ØAR and $ØBR) for subroutine A and locations 13R and 14R ($ØDR and $ØER) for subroutine B. The subroutine addresses default to an RTS.

CONTROL SEQUENCE SUMMARY

CHAR	FUNCTION
CTRL-A n	Select Character Set n
CTRL-B	Begin Block Display
CTRL-C	Carriage Return
CTRL-D	Delimit Block Display
CTRL-E	Clear to End of Line
CTRL-F	Clear to End of Screen
CTRL-G	
CTRL-H	
CTRL-I	Inverse Video
CTRL-J	
CTRL-K	Caps Lock *
CTRL-L	Lower Case
CTRL-M	
CTRL-N	Normal Video *
CTRL-O	Option Select →
CTRL-P	Clear Page
CTRL-Q	Home Cursor
CTRL-R	
CTRL-S	Shift
CTRL-T	
CTRL-U	
CTRL-V	Set Text Window (upper-left)
CTRL-W	Set Text Window (lower-right)
CTRL-X	
CTRL-Y	Set Text Window (full screen) *
CTRL-Z	Restore Default Parameters

OPTION	
CTRL-A	Primary is Page 1 *
CTRL-B	Primary is Page 2
CTRL-C	Complement
CTRL-D	Display Primary
CTRL-O	Overstrike
CTRL-P	Print *
CTRL-R	Reverse Overlay
CTRL-S	Scroll *
CTRL-T	Transparent Overlay
CTRL-W	Wrap
CTRL-Y	Call User Sub A
CTRL-Z	Call User Sub B

* Default Parameter

The default Character Set is ASCII.SET

DIFFERENCES FROM STANDARD TEXT DISPLAY

Although the Hi-Res Character Generator is intended to mirror the operation of the standard text display, there are several differences:

* The cursor is a flashing underscore character _ .

* The Applesoft command HOME and the monitor Call -936 will set the print position to the upper left corner of the text window, but they will not clear the screen; substitute CTRL-P .

* The Monitor commands CALL -958 and CALL -868 do not clear to end of screen and clear to end of line; substitute CTRL-F and CTRL-E .

* The Applesoft commands NORMAL, INVERSE, and FLASH do not control the Hi-Res video. Programs must write NORMAL video and use CTRL-N and CTRL-I to select normal and inverse video. Flashing video is not provided by the Character Generator.

* The HRCG uses the DOS input/output hooks, so it cannot be used with any other program that uses the hooks.

LEAVING THE HI-RES CHARACTER GENERATOR

To get out of the HRCG, press RESET, then type
FP
to reset HIMEM. Don't use the TEXT command: it will get you back to the text screen, but the HRCG will still be running: you won't have a cursor.

USING THE HRCG UNDER PROGRAM CONTROL

The Hi-Res Character Generator is intended to be used under the control of a "host" program, such as RIBBIT, although it can be run by itself. The HRCG program and the standard character set together require approximately 2.ØK bytes of memory. Each alternate character set will require an additional .75K bytes of memory. The program will run on a 32K system if the second hi-res graphics page is not used, and on a 48K system if the second page is used: on a 32K system, the second hi-res page is in the section of memory occupied by DOS, so writing to the second page would clobber DOS.

To make the HRCG as versatile as possible, it was written in relocatable form, so that the host program can put it into the most convenient place in memory. Before it can be used, HRCG must be relocated and initialized. The program is relocated by RLOAD, the relocating loader (described in its own chapter). Then it is initialized by a CALL to one of the two entry points. Since the Character Generator may be relocated to run at any address, entry points and other significant addresses within the Generator are given in relocatable form. The program LOADHRCG, which loads HRCG, should be studied if you want to write your own programs to use the Hi-Res

Character Generator. Absolute addresses are computed by adding the relocatable addresses to the relocation base returned from RLOAD. Relocatable addresses are denoted by the suffix R.

The program's main entry point is at the relocatable address ØR ($ØØR). When it is CALLed, the following initialization is performed:

1. The Character Generator is linked into the DOS input and output hooks.

2. All parameters are set to their default values.

3. The display is set to full screen hi-res graphics with the primary page displayed.

4. The screen is cleared and the title is displayed.

A second entry point, which performs only steps 1 and 2, is located at 3R ($Ø3R). This entry point is provided for applications that must not write on hi-res page 1. These applications must set the primary screen to page 2 before any data are written. For identification purposes, the Character Generator's version number is multiplied by 1Ø and stored at location 6R ($Ø6R).

CHAPTER 3
ANIMATRIX

Animatrix is an Applesoft program that lets you create and edit character sets for the Hi-Res Character Generator. Each of these character sets is a set of 7-by-8-dot images, which can be put anywhere on the Apple screen. Each character image comprises 8 rows of 7 dots: each row is represented by one byte, which comprises 7 bits for the individual dots and one dot for a row-shift.

Unlike the Hi-Res Character Generator, Animatrix is not intended to be used under the control of another program. It is used to create character sets before you run the HRCG.

STARTING UP

Make sure that DOS is booted and that the Applesoft Tool Kit diskette is in drive 1. Type
FP
to reset HIMEM, then type
RUN ANIMATRIX

When the program starts, it will print
CHARACTER SET TO EDIT?

If you wish to edit an existing character set, type in the name of the file containing the character set. If you wish to create a new character set, press RETURN. For now, respond by typing
ASCII.SET

The screen contains four areas of information, which will be blank when you start. Now type the letters of the alphabet, slowly, and they will appear on the screen, one by one. On the left side of the screen is a grid (green if you have color) containing an expanded image of the characters you typed. At the upper right is a box labeled 'Letters', showing the standard ASCII image of the characters you typed, as it would be printed by a dot-matrix printer. Below this is a box labeled 'Graphic', containing actual-size images of the characters you typed, as defined in the character set you just loaded. The images in the Letters box are for identification only: they will not change when you edit the images in the Graphic box and the expanded grid. Below the Letters box is the word <UPPER> or <lower>, which tells you whether the keyboard input is read as upper- or lowercase characters.

For example, if you are editing the ASCII.SET and you type
ABCDEFG
in uppercase mode, you will see the standard images of these letters in the 'Letters' box, and their graphic images in the Graphic box, exactly as they look when displayed, and expanded in the grid, in a more schematic form. You will notice some interesting differences

between the various forms. First compare the A in the Letters box to the A in the Graphic box: the first is house-shaped, and the second is more triangular. This is because Animatrix and the Hi-Res Character Generator can shift each of the horizontal lines making up a character, thus smoothing its contours. Here the second and fourth rows of dots are shifted one half-dot to the right. Now look at the expanded grid: the expanded A is more jagged-looking than the Graphic A, because while the Graphic box shows the half-dot shift graphically, the grid shows it schematically with a white square on the border between A and B. Showing the shift schematically makes character editing easier, because editing is all done by turning dots on and off, and because the white square makes it certain whether a row is shifted or not.

Now type

CTRL-Z

and you will be in lowercase mode, indicated by the <lower> below the Letters box. If you now type

abcdefg

you will see identifying images in the Letters box, and graphic images in the Graphic box and the expanded grid. If you wish to switch back to uppercase mode, type

CTRL-A

and keyboard input will be read as uppercase characters.

If you want to see the entire character set at once, typing

CTRL-D

gives you the character set in tabular form, with its name at the top, and five columns of characters. In each column the numbers on the left are the ASCII codes, the characters in the middle the standard ASCII characters, and the characters on the right the characters of the named character set, ASCII.SET. Only the numbers from 32 to 126 are used, as the other numbers are used for control characters. (Character 32 is the space.) This table is especially helpful when you are working with graphic character sets: if you want to edit a particular graphic character, but can't remember the standard character it corresponds to, just look in the table.

To restore the grid, press the space bar.

You can also get a summary of commands, by typing

CTRL-I

To restore the grid, press the space bar.

GETTING CHARACTERS ONTO THE SCREEN

To edit or create an image for a given character, first decide where you want it to go, then move the cursor in the Letters box to the square you want, by using the arrow keys and the RETURN key: the arrows will move the cursor left and right, and RETURN will move it to the beginning of the next line, just as on the normal Apple screen.

Now press the key that generates the character you want to define or change: for example, to edit the letter A, press the A key while in uppercase mode. The letter will appear as described above. When you type a character, its standard alphabetic image appears in the Letters box and its graphic image, if it is defined in this character set, appears in the Graphic box and on the grid. If you are defining a new character set, the Graphic box and the grid will both be empty, even when you have filled the Letters box. CTRL-A switches to uppercase input, and CTRL-Z switches to lowercase. The current case is always displayed below the Letters box.

The underscore, _ , backslash, \ , and left bracket, [, characters cannot normally be typed from the Apple keyboard, but can be input using the following control characters:

Underscore	-	CTRL-O
Backslash	\	CTRL-L
Left bracket	[	CTRL-K

The same control characters are used by the APA &Keys command.

EDITING CHARACTERS

Once you have entered the characters to be defined or changed, you can edit them using the paddles and the image on the grid. The paddles move a flashing cursor on the grid: paddle Ø controls horizontal motion; paddle 1 controls vertical motion. When the curser has been positioned over the desired dot in a character image, button Ø will turn the dot off and button 1 will turn the dot on. Each change will be shown both on the grid and in the Graphic box.

CTRL-T will toggle the shift bit of the row the grid cursor is in. When on, this bit shifts the row half a dot to the right, which changes the color of the dot if a color monitor is used. This shift can be used to smooth the contours of a letter. The shift will only be visible on the actual-size image at bottom right, not the expanded image on the expanded grid. On the grid, the shift will be indicated schematically by a white square on the right border of the letter image.

DUPLICATING CHARACTERS

If you wish two cells in the grid to have the same image, simply input the same character in the corresponding cells of the Letters box. From now on, until you change the characters in the Letters box, these cells, in each box, will behave the same. If one is undefined, the other will be undefined as well. If you define one, the other will be defined the same way. If you change one, the other will change the same way.

This feature is mainly useful in defining graphic character sets, which may repeat the same pattern in different parts of the image. If, for example, you define a character set consisting of several different running people, each several characters high, and the same character is used for the head of each, then you can change all the heads by changing one. Or one character can be used for a head turned toward the screen, and one for a profiled head, and each can be used with any of the runners. This shortcut can save time, and improve consistency.

If you wish to create several characters that are similar, differing by only a few dots, here's an easy way. Any undefined character can be made a duplicate of a defined one, so that both characters (in the Letters box) have the same images (in the Graphic box and the grid). To do this, pick a defined character (to be duplicated) and an undefined character (to duplicate the first). Let's say you have already defined O and P, and you wish to define Q and R. Input the first defined character, O, in the Letters box, put the cursor on it, then type
CTRL-S

You will be asked,

Same image for which character?

Now input the undefined character, Q. In the Letters box, the newly defined Q will replace the already defined O anywhere it occurs, and the cursor will move to the next space. In the Graphic box, all squares corresponding to the newly defined character Q will display the new image, which will be like that for O. Now you can repeat this process to make R like P. You can now edit the images for Q and R, to differentiate them from O and P. A duplicate image created with CTRL-S can be edited independently from its original: since they are linked to different characters, they will not change as one.

CTRL-S may only be used to define new characters: it will not change a character that has already been defined.

GRAPHIC CHARACTER SETS

To see what a graphic character set looks like, load FROGS1.SET and type in the following array of letters:

```
ABCDEQ
FGHIJKR
  LMNOP
!#  ABC
$%lmnoj
```

As you type in the letters, various creatures will appear on the screen.

LEAVING ANIMATRIX

To exit the program, press ESC. If you were editing an existing character set, you will be asked whether you wish to save the character set under its old filename. If you reply
Y
the old character set will be replaced by the new one. If you reply
N
or if you were creating a new character set, you will be asked for a new filename. When you type the new name and press RETURN, the new character set will be saved under the new name, and the old character set will be left undisturbed. If you press RETURN without typing a filename, the new character set will be erased as you leave the program.

At this point, you still have a chance to back out. After any of these options, you will then be asked whether you really want to quit. If you answer Y, you'll leave Animatrix; if you answer anything else, you'll return to the program, your current character set still in memory, and your old character sets unchanged.

If you leave Animatrix, you will still be in the Hi-Res Character Generator, and the character set you defined will be alternate character set 1.

To edit another character set, just type
FP
then RUN ANIMATRIX again.

SUMMARY OF ANIMATRIX COMMANDS

STRING INPUT COMMANDS

Left arrow	move cursor left
Right arrow	move cursor right
RETURN	go to next line
CTRL-A	shift to upper case
CTRL-K	left bracket, [
CTRL-L	backslash, \
CTRL-O	underscore, _
CTRL-Z	shift to lower case

CHARACTER EDITING COMMANDS

Paddle Ø	move grid cursor horizontally
Paddle 1	move grid cursor vertically
Button Ø	turn bit off
Button 1	turn bit on
CTRL-T	toggle high bit for half-dot shift
CTRL-S	same character image

OTHER COMMANDS

ESC	save character set and quit
CTRL-D	display entire character set
CTRL-I	display summary of instructions

CHAPTER 4

THE RELOCATING LOADER

The Relocating Loader lets you load Relocatable assembly-language modules (R files) from an Applesoft program, so that the program can run on Apples of different memory sizes. The Relocating Loader, which is made up of two programs, RLOAD and RBOOT, loads R files just below HIMEM, reducing HIMEM by the length of the modules loaded plus one byte (by length+1).

RESTRICTIONS

If your program uses string variables, use RBOOT and RLOAD before allocating or DIMensioning any strings, because RLOAD will be "hidden" between the memory locations containing your program lines and the locations containing your strings: if you allocate strings and then use RLOAD, RLOAD may land on top of your strings.

RBOOT

RBOOT is a small program that is BLOADed into addresses $2Ø8 through $3CF and invoked with a CALL 52Ø. Note that the usual D$ for the DOS BLOAD should be replaced with a CHR$(4), because you cannot allocate the string D$. RBOOT loads RLOAD above the end of APPLESOFT's variable table and sets the USR function to point to the entry point of RLOAD.

RLOAD

Once RLOAD has been set up, it can be used to load Relocatable modules. When accessed via the USR(Ø) function, RLOAD returns the starting address of the module. An "ON ERROR" return indicates that a problem was encountered during the attempted load. RBOOT and RLOAD both assume that the ON ERR statement is executed before calling them. The following example shows the syntax for RBOOT and RLOAD:

```
1ØØ  ADRS=Ø
11Ø  PRINT CHR$(4);"BLOAD RBOOT":CALL 52Ø
12Ø  ADRS=USR(Ø),"MYMODULE, S6, D1"
```

The slot and drive parameters are optional; the filename "MYMODULE" must refer to a Relocatable file.

The program LOADHRCG exemplifies the use of the Relocating Loader.

INDEX

Q

R

S

T

U

V

W

X

Y

Z

apple computer inc.

10260 Bandley Drive
Cupertino, California 95014
(408) 996-1010

030-0113-00

Addendum to the
Applesoft Tool Kit Manual

The information here supplements the information in the Applesoft Tool Kit Manual. When specific pages in that manual are relevant to the following points, the manual page number(s) are mentioned so that you can refer to the manual as you read this. You may wish to note some of these points in your manual.

page

5 The Note on this page warns you not to change the input/output hooks while the APA is in memory (by typing a PR# or IN # command or RUNning a program which does). Actually, there are times when the I/O hooks can be changed while the APA is loaded, depending upon which of the APA commands you are using. The only situations you MUST avoid are:

* Using a PR# command with either of the &Show and &Noshow commands.

* Performing an IN# command in conjunction with any of the &Auto, &Keys and &MAnual commands.

Remember that you must not change HIMEM or use the FP command while the APA in in memory.

1Ø The APA's &Xref command lists each variable followed by the numbers of the lines in which it appears. Unlike Applesoft's TRACE command, &Xref lists each line number only once for each variable, even if there are multiple statements in a single line which contain the same variable.

11 To leave the APA, press RESET. Then type
FP
to reset HIMEM.

11 It may be possible to have the APA in memory at the same time as another routine that uses the ampersand vector. To use such a routine without losing the APA, you must first perform a
CALL {address}
where {address} is the memory location of the beginning of that routine. Then the ampersand vector should be reset by the routine. The routine can then be accessed with the & function.

(continued on other side)

To return to the APA without losing the other routine, follow the same procedure for the APA. The beginning address of the APA would have to have been determined beforehand, immediately after you loaded the APA. To do this, you type
PRINT ADRS
right after loading the APA. The number you'd see then would be the correct beginning address of the APA for that session. Don't forget to determine this right after loading the APA if you plan to be using another routine which resets the ampersand vector.

13 The Hi-Res Character Generator can produce a number of character sets in both upper and lower case. The character sets Shadow.Set and Graphic.Set, however, do not have pre-defined lower case characters in the HRCG.

13 While the HRCG is in memory, do not use the FP command, change HIMEM, change the input/output hooks (with a PR# or IN # command) or RUN a program which does any of these things.

A final point about both the APA and the HRCG:

Before switching to Integer BASIC, you must leave the APA or the HRCG (if either is in memory). Press RESET, then type
INT