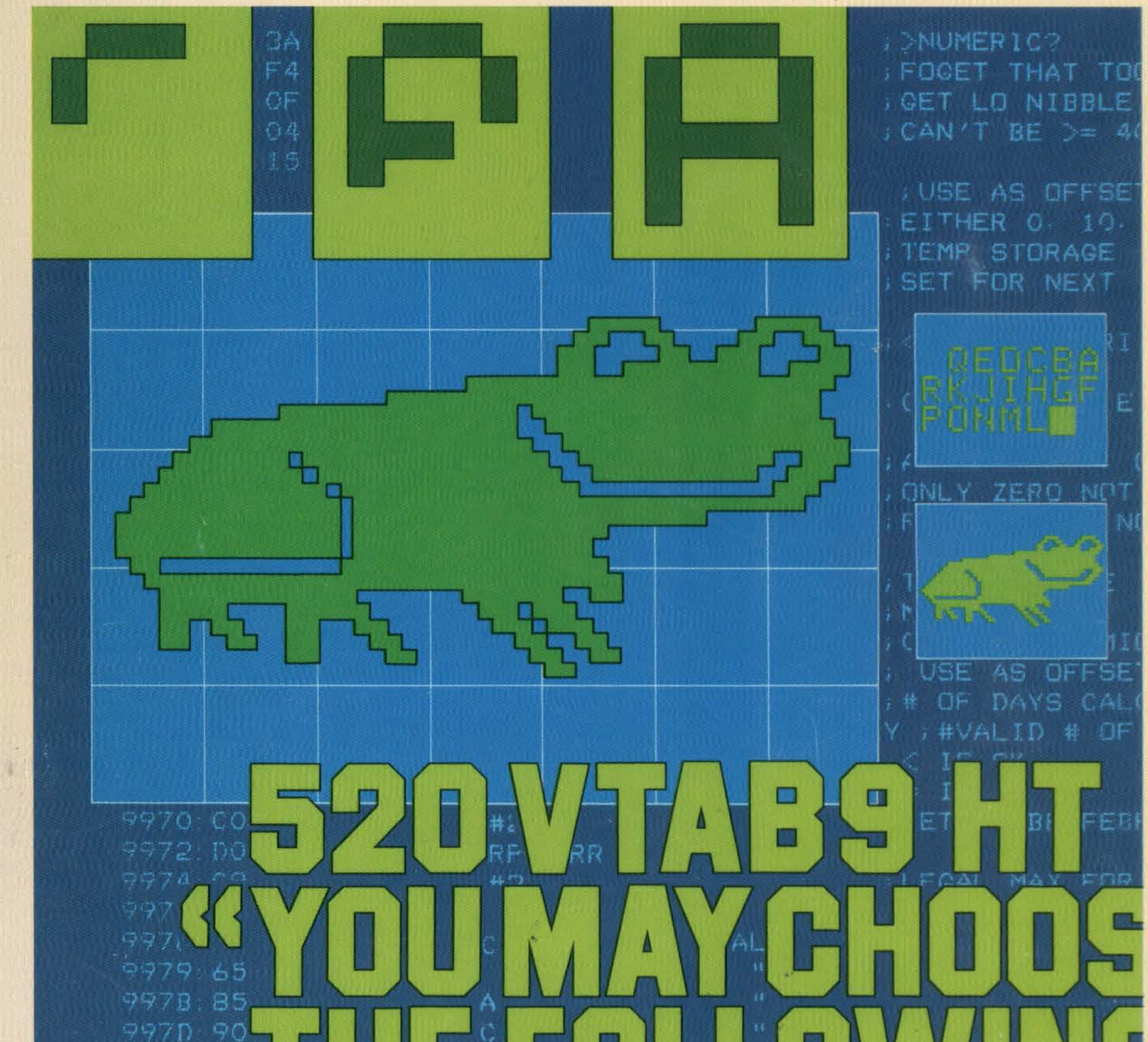
Weaver

Apple II

Applesoft Tool Kit



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. MAKES 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 INC. 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., ITS DISTRIBUTOR, OR ITS RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL APPLE COMPUTER INC. BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF APPLE COMPUTER INC. HAS 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.

©198Ø by APPLE COMPUTER INC.
1Ø26Ø Bandley Drive
Cupertino, California 95Ø14
(4Ø8) 996-1Ø1Ø

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

APPLE Product #A2LØØ38
(Ø3Ø-Ø113-ØØ)

Apple II

Applesoft Tool Kit

TABLE OF CONTENTS

INTRODUCTION

The Applesoft Tool Kit is designed to help you use the capabilities of the Apple II. The Applesoft Programmer's Assistant gives you the same facilities in Applesoft as you have in Integer BASIC: merging, renumbering, and automatic line numbering during program entry. Other programs make it easier to produce reliable and interesting interactive programs, such as games and educational programs. Animatrix and the High-Resolution Character Generator simplify graphics: the former lets the user define graphic character sets, and the latter makes it easier to display them. The Relocating Loader lets the user write programs that can run on Apples with various amounts of memory, by making assembly-language subroutines relocatable.

The Applesoft Tool Kit also includes three programs, RIBBIT, SKYLAB, and MAXWELL, which demonstrate the use of the High Resolution Character Generator and the Relocating Loader.

The Applesoft Tool Kit is intended for experienced users who are familiar with Applesoft II, DOS 3.3, and in some cases, 65Ø2 assembly language. It is not mainly intended for the novice user, although some programs can be used by novices--the games can be played by children. The DOS Manual and the Applesoft Tutorial or Reference Manual should be read before using the Applesoft Tool Kit.

The Applesoft Tool Kit is designed to use DOS 3.3. It will not run under previous versions of DOS. If you have Applesoft ROMs in your Apple (that is, if you have an Apple II Plus or an Apple II with the Applesoft Firmware Card), you can boot DOS 3.3 from the Tool Kit diskette and run its programs by simply inserting it into drive 1 and following the normal boot procedure. If you have an Apple II with the Language System, you must first boot the DOS 3.3 System Master, to load Applesoft into the Language Card.

The programs described in this manual will run on an Apple II Plus or an Apple II with Applesoft Card or Language System, and at least one Disk II drive and controller. The 16-sector disk controller PROMs, P5A and P6A (or 341-ØØ27-xx and 341-ØØ28-xx, where xx is the batch number), must be installed in each disk controller.

The Hi-Res Character Generator requires either 32K or 48K of memory: 32K if only the first hi-res graphics page is used, 48K if the second page is used. At least 32K of memory is required for the Assembler/Editor and the Applesoft Programmer's Assistant; 48K is required for the other programs on the Tool Kit diskette.

The Assembler/Editor (described in its own manual) does not require Applesoft, and can run on an Apple II without Applesoft Card or Language System, if the Apple has one or more disk drives, with PROMs P5A and P6A in each disk controller, and at least 32K of memory.

CHAPTER 1

THE APPLESOFT PROGRAMMER'S ASSISTANT

The Applesoft Programmer′s Assistant (APA) does several things to help you write and change Applesoft programs. It numbers lines automatically as you enter a program, renumbers and merges programs at will, measures the length of a program, and deletes remarks from a program in its final form. It also shows you the control characters in a file, lets you type special characters from the keyboard, and cross-references programs.

STARTING UP

To use this program, make sure DOS is running, and that the Applesoft Tool Kit Diskette is in drive 1, then type

RUN LOADAPA

When you press RETURN, the drive will whir and zick, and the LOADAPA program will relocate, load, and initialize the APA. If you want, you can now LIST the program, to see how it works.

USING THE APA

Once you have examined the LOADAPA program, RUN it. LOADAPA will LOAD APA just below HIMEM, then reset HIMEM just below APA, thereby hiding APA so it can′t get erased accidentally. After the APA is loaded and hidden, you will see a title, a copyright message, and a list of commands. These are shown below in a slightly different form from what you′ll see on the screen, in order to clarify their syntax. Each of the items in angle brackets, like <start>, is an integer parameter; anything in square brackets, like [, <last>], is optional. Optional items may be nested: you can omit the commas and parameters inside each set of brackets, matching the brackets from the innermost outward. Neither angle brackets nor square brackets should be typed at the keyboard--only the ampersand, the uppercase letter or letters of the command name, the parameters, and the commas between the parameters. The spaces in the syntax diagrams are optional, and are the only ones allowed--don′t space after the ampersand.

&Renumber [<start>] [, [<inc>] [, [<first>] [, <last>]]]

&Hold

&Merge

&Length

&Compress

&Show

&Noshow

&Auto <start>[, <inc>]

&MAnual

&Keys

&Xref

These commands are discussed in detail below. The lower-case letters in the command name may be omitted, as may the numeric parameters in square brackets: e.g., [<inc>]. These parameters (following &Renumber and &Auto) are integers between 1 and 63999, and are identified by position: if you leave one out, put in the comma following it so the APA will know it was left out.

THE COMMANDS

&Auto <start>[, <inc>]

This command gives you automatic line numbering. You can use it while entering a program, as the APA "hides" at the top of memory, where it can not clobber, or be clobbered by, your program. If your program is extremely large, you may need the space taken up by the APA, but this will rarely be a problem.

To write a program starting with line number 1ØØ, simply type
&A 1ØØ
and press RETURN. Now press the space bar, and a line number will appear. Type your statement, press RETURN, and press the space bar: the next line number, 11Ø, will appear. Unless you state otherwise, the increment <inc> will be 1Ø.

If you wish to leave a line number unused, press RETURN, then the space bar: the next line number will appear, and no statement will be entered under the previous one.

If you change your mind about a statement while you are typing it, press CTRL-X: the same number will reappear, and no statement will be entered.

Since a line number appears only if you press the space bar after the prompt ( ]), you can type a run-time command any time you see the prompt, as long as you don't begin it with a space. This means you can RUN or LIST your program, or SAVE it and LOAD another, just as if the APA weren't there.

Note: It is possible to clobber the APA with run-time commands. Using the FP command will do it, as will changing HIMEM or the input/output hooks. For this reason, when the APA is loaded, avoid RUNning any program that changes HIMEM or the I/O hooks, or pressing RESET.

&MAnual

To turn off automatic line numbering type

```
&MA
```

and press RETURN. &MAnual is the default when you load the APA: you have to ask for automatic line numbering to get it.

&Renumber [<start>] [, [<inc>] [, [<first>] [, <last>]]]

The &Renumber command will renumber any part of your program, with any starting line number and increment, as long as this renumbering will not cause interleaving of lines or duplication of line numbers. Interleaving or duplication would occur if a line range were renumbered so that its new numbers overlapped the numbers of an existing line range. If an &R command would cause this to happen, the command will not be executed.

Let's start with an example. Enter this simple program at the keyboard, and SAVE it as, say, SAMPLE:

```
1Ø PRINT "S"
2Ø PRINT "A"
3Ø PRINT "M"
4Ø PRINT "P"
5Ø PRINT "L"
6Ø PRINT "E"
```

To renumber this program, type

```
&R
```

Your whole program will be renumbered, starting with number 1ØØ and incrementing by 1Ø.

```
1ØØ PRINT "S"
11Ø PRINT "A"
12Ø PRINT "M"
13Ø PRINT "P"
14Ø PRINT "L"
15Ø PRINT "E"
```

You can specify a different starting line number <start> and increment <inc> by typing, say,

```
&R 1ØØØ, 5Ø
```

where the starting number is 1ØØØ and the increment is 5Ø. This will give you

```
1ØØØ PRINT "S"
1Ø5Ø PRINT "A"
11ØØ PRINT "M"
115Ø PRINT "P"
12ØØ PRINT "L"
125Ø PRINT "E"
```

If you wish to renumber only part of your program, you can specify the <first> and <last> old line numbers to be changed. For example, if you reLOAD SAMPLE, then type

```
&R 1ØØ, 1Ø, 3Ø, 4Ø
```

--where the starting number is 1ØØ, the increment is 1Ø, and the first and last old line numbers are 3Ø and 4Ø, respectively--you will get

```
1Ø PRINT "S"
2Ø PRINT "A"
5Ø PRINT "L"
6Ø PRINT "E"
1ØØ PRINT "M"
11Ø PRINT "P"
```

Only the lines previously numbered 3Ø and 4Ø have been renumbered. Notice that the lines have been reordered to keep the line numbers in order. This feature can be used for such purposes as moving subroutines to the end of your program.

The command above is equivalent to

```
&R , , 3Ø, 4Ø
```

The APA uses the default value whenever it sees a comma without a parameter before it. The default values are

<start>	1ØØ
<inc>	1Ø
<first>	Ø
<last>	63999

The Renumber command will not let you use parameters that would give two lines the same number or interleave two sets of lines. In this situation, you will get an error message. For example, LOAD SAMPLE and type

```
&R 1Ø, 5, 4Ø
```

You will get the message

```
INTERLEAVED OR DUPLICATE LINE NUMBER
```

because this command would not only put line 5Ø between lines 1Ø and 2Ø but also put lines 4Ø and 6Ø on top of lines 1Ø and 2Ø. In this respect, the APA differs from certain other renumbering programs, which would merrily scramble your file when given such a command.

The Renumber command will change references from one line to another, such as GOTOs, GOSUBs, ONERRs, and the like. So if the program ran correctly before renumbering, it will run correctly after, unless you changed the order of the lines. Renumber will not change line references in REM statements, so check these yourself.

&Hold

This command hides your program above HIMEM so it can't be erased, and lets you LOAD another program into the area your program just left. To use it, make sure you have a program in memory, then type

```
&H
```

Now try to LIST your program. It will be gone. To bring it back, read on.

&Merge

To get your program out of the Hold area, type

```
&M
```

LIST it, and it will reappear.

You can use this command to merge two programs. Let's start with our SAMPLE program. First LOAD and LIST it on the screen:

```
1Ø PRINT "S"
2Ø PRINT "A"
3Ø PRINT "M"
4Ø PRINT "P"
5Ø PRINT "L"
6Ø PRINT "E"
```

Now delete lines 1Ø, 3Ø, and 5Ø by typing the line numbers, followed by RETURNs. When you LIST the program, you will have

```
2Ø PRINT "A"
4Ø PRINT "P"
6Ø PRINT "E"
```

SAVE this as APE, or whatever. Now type

```
&H
```

and these lines will be hidden.

Now reLOAD SAMPLE, and delete lines 2Ø, 4Ø, and 6Ø, as above. This will LIST as

```
1Ø PRINT "S"
3Ø PRINT "M"
5Ø PRINT "L"
```

SAVE this as SML, or whatever. Now type

```
&M
```

and LIST the result:

```
1Ø PRINT "S"
2Ø PRINT "A"
3Ø PRINT "M"
4Ø PRINT "P"
5Ø PRINT "L"
6Ø PRINT "E"
```

Presto! The program you carved up so diligently is now whole again, and sorted by line number. If you had renumbered SML before merging, for example, by typing

```
&R , 2ØØ
```

so that SML looked like this

```
1ØØ PRINT "S"
3ØØ PRINT "M"
5ØØ PRINT "L"
```

you will get different results. Type

```
&M
```

and you will get

```
2Ø PRINT "A"
4Ø PRINT "P"
6Ø PRINT "E"
1ØØ PRINT "S"
3ØØ PRINT "M"
5ØØ PRINT "L"
```

This command, unlike the Renumber command, can create duplicate line numbers. For example, put APE on Hold, then LOAD SML. Renumber it

```
&R 2Ø, 1Ø
```

and you will have, on LISTing,

```
2Ø PRINT "S"
3Ø PRINT "M"
4Ø PRINT "L"
```

Notice that both APE and the new SML have lines numbered 2Ø and 4Ø. Now type

```
&M
```

and you will get this message

```
DUPLICATE LINE NUMBER
2Ø
4Ø

CONTINUE?
```

If you do, whenever two lines have the same number, the line from the Hold area will be chosen, and the other line with the same number will be deleted. You will now have

```
2Ø PRINT "A"
3Ø PRINT "M"
4Ø PRINT "P"
6Ø PRINT "E"
```

which may be what you had in mind, or may not.

With this command, you can perform wonders, and save yourself much time. You can also wreak considerable havoc, so it is advisable to SAVE each of the programs to be merged, as "all the king's horses and all the king's men ..."

&Compress

This command will remove REMarks from the program in memory. The Compress command will let you maintain two versions of a program: one documented and one compressed. The first will be easier to maintain, and the second will be smaller and run faster. To use this feature, make sure you have SAVEd the current version of your documented program, then type

```
&C
```

The program in memory will be compressed, and can then be SAVEd separately. If you later revise the program, you can change the documented version and make a new compressed version from it.

&Length

This command gives the Length of the program in memory. To use it, type

```
&L
```

and you will get the program's length in bytes, in both decimal and hexadecimal form.

&Show

This command makes the control characters in your program visible. Type

```
&S
```

and control characters will be displayed in inverse video.

&Noshow

This command makes control characters invisible. Type

```
&N
```

and control characters will not be displayed.

&Keys

This command lets you type three additional characters at the keyboard: the underscore, _ ; the backslash, \ ; and the left bracket, [. To use this command, type

```
&K
```

From now on, you can input these characters by typing the following control characters:

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

To cancel the &Keys feature after you no longer need it, type
&MA

Note: You can't use both this feature and the &Auto feature at the same time. If you type &K when &Auto is on, you will turn &Auto off and &Keys on; if you type &A when &Keys is on, you will turn &Keys off and &Auto on.

&Xref

The &Xref command gives you a cross-reference listing of an Applesoft program, listing each variable, followed by the numbers of the lines in which it appears. To use it, LOAD the program you wish to cross-reference, then type
&X
and the variables and their line numbers will be listed on the screen. If your cross-reference listing is long, you can interrupt it by typing
CTRL-S
if you have the Autostart ROM in your Apple.

To see how this command works, type
FP
and RUN LOADAPA, then type
&X

You will get this table on the screen.

```
      AD 13Ø, 16Ø, 18Ø
```

showing all the lines where AD, the LOADAPA program's only variable appears. To see how &Xref works on a large program, LOAD the program RIBBIT. When the program is in memory, LIST it and watch its many lines go by on the screen. Now type
&X

You will get a very long cross-reference table, which can be interrupted by the CTRL-S command. Note that all variable names will be shortened to two characters if they are longer than two, as Applesoft only distinguishes the first two characters of a variable name. In the case of a string, these will be followed by a dollar sign, $; in the case of an array, an open parenthesis, (; in the case of a string array, both, $(. Names of integer variables will be followed by a percent sign, % ; integer arrays will be followed by a percent sign and an open parenthesis, %(. For example, the cross-reference table for RIBBIT begins with the variables

A$	(which is a string variable),
AD	(which is a real variable),
BC	(which is a real variable),
BD	(which is a real variable),
BS	(which is a real variable),
BU$(	(which is a string array variable),

each variable name followed by a set of line numbers.

USING THE APA UNDER PROGRAM CONTROL

You can also RUN the APA program from inside your own program. To do this, copy LOADAPA into your program, using the Hold and Merge commands, and the Renumber command if necessary. You want to be sure, when you do this, that LOADAPA does not interleave with a section of your program.

To see what LOADAPA will do, LOAD LOADAPA without RUNning it, then LIST it on the Apple screen.

If you don't want your program to print the title, change line 16Ø to

```
16Ø ADRS = USR (Ø) + 3
```

This will cause the program to start at its second entry point, which skips the title. You can get rid of the other displays as well, by deleting the PRINT statements, except 14Ø PRINT CHR$ (4),"BLOAD RBOOT" which does part of the relocating, and 29Ø PRINT "UNABLE TO LOAD" which is an error message. If line 14Ø is deleted, the Relocating Loader will not be loaded, and if line 29Ø is deleted, you may not find out if something goes wrong. By leaving 29Ø in, you can be sure that "No news is good news."

CHAPTER 2
HI-RES CHARACTER GENERATOR

The Hi-Res Character Generator (HRCG) is a set of assembly-language subroutines for displaying text on the high-resolution graphics screen. It serves the same function as the Apple's hardware character generator (described in the section The Video Generator in the Apple II Reference Manual). It turns a series of ASCII character codes into a bit-by-bit map of the Apple screen: each one-byte ASCII code produces a 7-wide-by-8-high dot image on the high-resolution screen, covering an area equal to two low-resolution blocks stacked vertically. Each dot will be on (white) or off (black), and the dots in each line of a character-image can be shifted sideways by half a dot to produce smoother contours. The Hi-Res Character Generator can produce the same character set as the hardware character generator, and many others besides: foreign alphabets and decorative alphabets, both with upper- and lowercase, and graphic alphabets for animation, which carve up a large image into 7x8 blocks which can be displayed anywhere on the screen.

With its default settings, the HRCG behaves much like the standard text display. The text window fills the screen, the screen scrolls, and Applesoft's VTAB and HTAB can be used to control the print position.

Beyond its ability to display upper- and lowercase text, the Hi-Res Character Generator provides several additional display capabilities:

* Intermixing of text and high-resolution graphics

* Normal and Inverse video

* Alternate character sets for user-defined characters

* The ability to display graphic character sets for animation

* The ability to write text over an existing background

* Scrolling or text wrap-around within the text window

* Automatic downshifting of alphabetic characters for displaying lowercase text

The Hi-Res Character Generator is stored under the filename HRCG on the Applesoft Tool Kit diskette. The diskette also includes the program LOADHRCG, which loads and initializes the HRCG program, as well as a number of character-set files. These include ASCII.SET, a standard ASCII character set; GREEK.SET, a Greek character set; many decorative alphabets; and several graphic character sets for animation. The diskette also contains the programs RIBBIT, SKYLAB, and MAXWELL, which show how the HRCG is used for animation.

LOADING CHARACTER SETS

To see how the HRCG works, we will use the program LOADHRCG to load and initialize it. Normally the HRCG would be loaded by a "host" program, like RIBBIT, which will provide the HRCG with text to display. How to use the HRCG under program control will be explained later in this chapter, but for now it will be simpler to control it directly by keyboard commands.

Before using the HRCG, we must load the desired character sets from diskette. To do this, first make sure you have DOS booted, then CATALOG the Applesoft Tool Kit diskette and write down the filenames of the character sets you want. Now reset HIMEM by typing
FP
then load the HRCG by typing
RUN LOADHRCG

(Note: It is always wise to use the FP command before loading the Character Generator, especially if you just used the Character Generator, then left the program. The Character Generator changes the value of HIMEM, and FP resets HIMEM to its default value. If you run LOADHRCG right after booting the system, FP is unnecessary.)

You will be asked how many character sets you wish to load. Answer by typing Ø if you will only use the standard set, or by typing a number from 1 to 9 if you want to load additional ones; then press RETURN. This will make the program allocate enough space in memory to hold your character sets. When the program is loaded, you will be asked for the name of the first alternate character set. Type the filename of this set, then press RETURN. Repeat until you have loaded all the character sets you want.

To see the alternate character sets, you can type
CTRL-A
followed by the number, n, from 1 to 9, of the alternate character set. Ignore the SYNTAX ERROR message, which results from the fact that the command to the HRCG is not a valid BASIC statement. You can suppress the error message by typing
CTRL-A n CTRL-X
(with no spaces). If you now LIST the program, LOADHRCG, it will come out in the new character set. To get back to the standard character set, type
CTRL-A Ø CTRL-X
(with no space). If you LIST the program, it will come out in the normal letters, but it will scroll differently: it will move up one bit at a time, rather than one byte; or one scan line, rather than one line of text. It moves at about the same rate, but in smaller jumps. This quirk, which derives from the bit-by-bit way in which the characters are produced, is a good way to recognize the Hi-Res Character Generator in operation. Each line of text is created one scan line at a time, rather than one character block at a time.

You can do this with any of the character sets you have loaded, to see what they look like. If you wish to change any of them, use ANIMATRIX, as described in its own chapter.

You may be amused by the results if you use one of the graphic character sets, say FROGS1.SET or SKY1.SET, to LIST the program. As always, typing

CTRL-A Ø CTRL-X

will get you back to the normal character set. If you press RESET, you will leave the HRCG entirely, and erase the character sets you have loaded. After pressing RESET, remember to use the FP command to reset HIMEM: if you RUN LOADHRCG repeatedly without resetting HIMEM, you will eventually run out of memory.

THE COMMANDS

The Hi-Res Character Generator's display format is controlled by PRINTing control sequences of one or two ASCII characters. The control sequences are described here, with a summary table at the end of the section. Most control sequences consist of a single ASCII control character, but some sequences require two control characters, and one requires a control character and a digit. In the descriptions below, control characters are separated by spaces for clarity. These spaces should not be typed. Default parameters are starred.

CTRL-A n   Select Character Set n

This control sequence selects character set n, where n is an ASCII digit. Character set Ø is the standard ASCII character set, comprising the 96 printing ASCII characters. You may also define up to nine alternate character sets, each with 96 additional symbols. Your character sets must reside in a contiguous block of memory. Before an alternate character set can be selected, its base address must be POKEd (low byte first) into the HRCG locations 7R and 8R ($Ø7R and $Ø8R). LOADHRCG takes care of this POKEing, so if you have run LOADHRCG, you can use the CTRL-A n command without further ado.

CTRL-N   Normal Video Display *

CTRL-I   Inverse Video Display

Normal video displays white characters on a black background. Inverse video displays black characters on a white background.

CTRL-O CTRL-P Print *

CTRL-O CTRL-O Overstrike

CTRL-O CTRL-C Complement

CTRL-O CTRL-T Transparent Overlay

CTRL-O CTRL-R Reverse Overlay

In Print mode, the printed character (white on black or black on white, depending on Normal/Inverse mode) replaces the current display. In Overstrike mode, the printed character (white or black, depending on Normal/Inverse) overlays the existing display. In Complement mode with Normal video, the character portion of the existing display is complemented; with Inverse video, the background is complemented. In Transparent mode, the character is overlaid on the secondary screen and the result transferred to the primary screen. In Reverse mode with Normal video, the character portion of the secondary screen is complemented and the result transferred to the primary screen; with Inverse video, the background is complemented and the result transferred.

CTRL-O CTRL-A Primary Screen is Hi-Res Page 1 *

CTRL-O CTRL-B Primary Screen is Hi-Res Page 2

The HRCG always writes on the primary screen. This parameter specifies which Hi-Res graphics page will be used as the primary screen. Note that changing the primary screen does not automatically change which Hi-Res graphics page is being displayed.

CTRL-O CTRL-D Display Primary Screen

This parameter sets the display to the primary screen.

CTRL-O CTRL-S Scroll Text Window *

CTRL-O CTRL-W Wrap Text Window

This parameter controls the Character Generator's action when it reaches the bottom of the text window. When scrolling is selected, the Character Generator moves the current display up one line and replaces the bottom line with blanks. When wraparound is selected, it simply moves the cursor to the top line of the text window.

CTRL-Q Home Cursor

Normally, CTRL-Q moves the cursor to the upper left corner of the text window. If the Character Generator is operating in Block Display mode, however, the cursor is moved to the upper left corner of the block.

CTRL-C Carriage Return

Normally, CTRL-C operates like a carriage return, moving the cursor to the left side of the text window and down one line. If the Character Generator is operating in Block Display mode, however, the cursor is moved to the left side of the Block and down one line.

CTRL-P Clear Page

CTRL-E Clear to End of Line

CTRL-F Clear to End of Screen

These control characters serve the same functions as the escape sequences ESC @, ESC E, and ESC F, or the Monitor commands CALL -936, CALL -868, and CALL -958.

CTRL-Y Set Text Window (full screen) *

CTRL-V Set Text Window (upper left)

CTRL-W Set Text Window (lower right)

These control characters are used to set the text window. CTRL-Y sets the text window to the full screen. With the window set to the full screen, CTRL-V sets the upper left corner of the text window to the current cursor position. CTRL-W sets the lower right corner of the text window to the current cursor position.