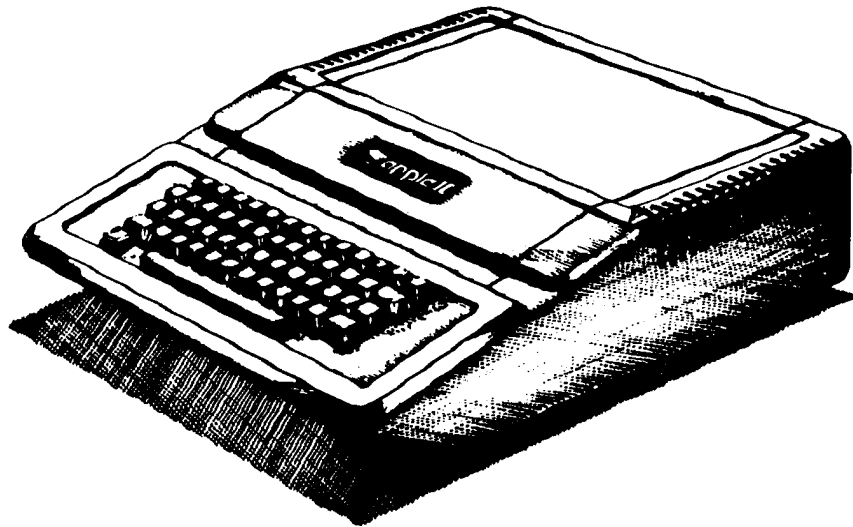


Apple 2 Computer Technical Information



Apple II Computer Family Information

AppleSoft BASIC Info :
History by Steven Weyhrich (Jan. 1992)

Document # **42**

Ex Libris David T. Craig

Apple II Computer AppleSoft BASIC Information
 Steven Weyhrich -- 22 Jan 92 -- File AH.16

APPLESOFT I

Although Wozniak had written some floating point routines into the Integer Basic ROM, Apple II users needed a version of Basic that would make floating point math easier to do, particularly for business use (where the number to the right of the decimal point is as important as the one to left). Apple decided to license a 6502 version of a floating point BASIC from Microsoft Corporation. Back in 1977, Microsoft was producing BASIC interpreters for nearly every microcomputer that was produced. The version Apple purchased was almost identical to the MITS extended BASIC that Microsoft had previously written for the Altair 8800.<4>,<5>

This BASIC was named "Applesoft", and was released in November of 1977 on cassette. It was loaded as a 10K program that looked to the computer just like an Integer BASIC program, though only a small part of it really was. To make it easy to load and start from cassette, the Applesoft interpreter was attached to the end of a short Integer BASIC program. When the Integer program was run, it poked some values into memory and jumped to the start of the machine language section, which relocated the Applesoft interpreter to the lower part of memory (at \$800), just after the memory that held the screen display.

Using this version of Applesoft (which later became known as Applesoft I) could be frustrating. It took several minutes to load from the cassette tape, and it was not dependable. If the wrong key was pressed while entering or running an Applesoft program, the program that was being run could be wiped out, and the Applesoft interpreter itself would have to be reloaded from cassette. However, few users knew how to make use of the floating point routines that Wozniak had written into the Integer ROM, so this unreliable Applesoft BASIC became the only practical means of doing floating point math on the Apple II.

Aside from the reliability issue, another difficulty with Applesoft involved hi-resolution graphics. Although the Apple II was capable of displaying it, the Applesoft interpreter extended up into the memory used by the hi-res screen, and so prevented its use. Furthermore, this early version had no built-in commands to manage hi-res graphics.<5>

Applesoft I came with a manual that was 8 1/2 inches by 11 inches in size, and sported a blue cover with square glued binding.<6> This came to be known as the "blue book" (recall that the reference book for the computer itself was affectionately known as the "red book"). When starting the interpreter after loading it from the cassette, a screen was display announcing that Applesoft was copyright 1977 by Apple and Microsoft. It then asked the user for the memory size of his computer, and gave options of allowing either LET and REM statements OR the use of lo-res graphics. The names of the lo-res graphics commands were very different from those that existed in Integer BASIC (and in the later versions of Applesoft). The commands were:

PLTG = Go to lo-res graphics mode
 TEX = Go to text mode
 PLTC N = Set color to N (0-15)
 PLTP X,Y = Plot square at X,Y
 PLTH X1,X2,Y = Plot horizontal line from X1 to X2 at Y

Apple II Computer AppleSoft BASIC Information 1
 Steven Weyhrich - 22 Jan 92 - File AH.16

PLTV Y1,Y2,X = Plot vertical line from Y1 to Y2 at X

There was a note about these commands in the reference card included with Applesoft I that warned about using graphics coordinates only between 0 and 39, or a program could "self-destruct". Apparently it lacked the error checking that could prevent the plotting of lines from spilling over into the text of the Applesoft program itself.<6>,<7>

The A.P.P.L.E. user group published a patch in 1978 that allowed programmers to avoid the question about using LET and REM statements versus lo-res graphics, and use the graphics only. The author of the patch pointed out that the LET statements were not necessary ("A = 3" worked just as well as "LET A = 3"). The REMark statements could be simulated by putting them at the end of a GOTO line (where they were ignored by the interpreter), and the GOTO could just jump to the following line:

```
530 GOTO 540: REM LINE 540 SETS VARIABLE N.
540 N = 2
```

Additional patches were made available for some of the other bugs found in Applesoft I.<8>

APPLESOFT II

In spring 1978, Randy Wigginton and some others at Apple made some needed revisions to Applesoft. Using a cross-assembler running on a North Star Horizon (Z-80) microcomputer, they fixed the known bugs and added other commands to control features unique to the Apple II. These commands included the ones needed to draw and manipulate hi-res graphics. Also, the lo-res graphics commands were renamed to be more consistent with the equivalent commands in Integer BASIC (GR, HLIN, VLIN, etc.) This version was called "Applesoft II", and eventually it was available in five forms: Cassette RAM and Diskette RAM (which loaded to the same memory locations that interfered with hi-res graphics as did Applesoft I), Firmware card ROM, Language card RAM, and finally main board ROM (in the Apple II Plus).

When Applesoft II was started up from cassette or diskette versions, the display screen now showed a copyright date of 1978 by Apple Computer, Inc., and 1976 by Microsoft (which may be either their copyright date for the original Microsoft BASIC, or possibly for Microsoft's first 6502 version).<6> This RAM version of Applesoft II used memory from \$800-\$2FFF, and the Applesoft BASIC program itself was loaded beginning at \$3000. When the versions that came on ROM and for the Language Card RAM were released, the BASIC program could load at \$800, and much more memory was available for it. Some of this extra space (in high memory) was reclaimed by DOS when the Disk II was released, however.<5>

Applesoft in the original IIe was unchanged from the II Plus version. When the IIc was introduced in 1984, however, Apple programmers had cautiously made a few useful changes to the language:

- o Input processing was changed to allow lowercase entry of Applesoft commands (they were translated into uppercase)
- o Screen output commands (PRINT, TAB, HTAB, etc.) were modified to more properly handle the 80-column screen
- o Program lines (when LISTed) were changed to begin in column 2, making screen editing easier

- o All of the cassette tape routines (LOAD, SAVE, SHLOAD, STORE, and RECALL) were removed, since the hardware did not support cassette I/O. The keywords were still in the token table, but now pointed to the same memory vector as the ampersand("&") command.
- o Patches were made to the lo-res graphics commands (GR, HLIN, VLIN, PLOT, and SCRN) to work with double lo-res graphics. However, a bug was introduced that allowed PLOTting vertically to areas outside of the double lo-res graphics screen, which would land right in the beginning of the \$800 space where the Applesoft program text was located (similar to the "plot" bug in Applesoft I).

When the Apple IIe Enhanced ROMs were made available, Applesoft in those ROMs had undergone some similar modifications. All the above IIC changes were added, with the exception that double lo-res graphics capability was NOT added (lack of ROM space), and the cassette I/O commands were NOT removed (since the cassette input and output port was still present).

The version of Applesoft on the Apple IIGS closely resembled the Apple IIC variant, the only exception being a fix of the double lo-res PLOTting bug. However, a bug in the SCRN function that applied to double lo-res mode was NOT fixed. No changes to Applesoft from the IIC version appeared in the Apple IIC Plus.<9>

The manuals written for Applesoft II were far more comprehensive than either the older "Blue book" or the Integer BASIC manual. It gave not only programming examples for each of the commands, but included much more information about the various ways in which each Applesoft statement could be used. It also mentioned some of the differences between Applesoft and Integer (for those who wanted to convert their older programs), and gave a little information about the internals of Applesoft to aid in creating machine language additions to the language. Curiously, the manuals that have been reprinted even as late as 1990 by Addison-Wesley have included an odd cautionary note to programmers. In a section in the index about "reserved words" (words reserved as Applesoft commands), it advises against using "XPLOT" as a variable name, stating that "it is a reserved word that does not correspond to a current Applesoft statement." What is apparently meant by this comment is that at one time Apple intended to extend the language and add another command "XPLOT" to it, probably working with HPLOT in the same way that XDRAW complements DRAW in doing hi-res graphics. Examination of the command table within the Applesoft interpreter shows there is NO entry labeled "XPLOT", and a disassembly of the interpreter shows NO preliminary code to support the command. Somehow this precaution persisted to the present day and has never been removed, even though it is extremely unlikely that Applesoft will ever be upgraded.<10>

Particularly helpful for programmers was the foresight to include a simple extension called the "ampersand hook". If Applesoft came across the "&" symbol while interpreting a line, it jumped to a known location in memory and left it to the programmer to insert the correct code to add a machine language extension to the language. With the publication of important information about the internals of Applesoft in 1980, assembly language programmers could now add statements to do things that could not be done with the language as it was originally created. Music, extended graphics, IF-THEN-ELSE logic, and even the missing "XPLOT" command could be added to the language. The only limits were the author's imagination (and available memory).

The importance of Applesoft as an influence to productivity on the

Apple II cannot be overstated. Since the release of the Apple II Plus in 1979, every variety of Apple II has contained Applesoft in virtually an unchanged form. This has made it possible for anybody to write programs that ALL other Apple II users will be able to use, since the language does not have to be purchased or added. If there were thousands of Integer BASIC programs from the two years when Integer Apple II's were produced exclusively, there are hundreds of thousands of Applesoft programs that appeared over that subsequent thirteen years. Even today, it is not uncommon for an applications program to include a configuration module written in Applesoft using the disk commands available with BASIC.SYSTEM in ProDOS. It is often faster to write such a program in BASIC, and the author knows without a doubt that his customer will be able to run it.

APPLESOFT 3 (?)

In 1979 there were rumors at the West Coast Computer Faire about an enhancement to Applesoft II that was in the works at Apple. It would possibly be called Applesoft 3, and would be as much of an enhancement over Applesoft II as that version was to Applesoft I. Supposedly it was intended to merge DOS and BASIC, and would include such powerful functions as IF-THEN-ELSE, PRINT USING, WINDOW, and VIEW PORT. It was predicted to be a RAM version only, and would be about 24K in size. Knowing the events that actually followed, this rumored BASIC was probably the "Business Basic" released with the Apple III, rather than an enhancement for the Apple II.<11>

NOTES

- <4> Chien, Philip. "The First Ten Years: A Look Back", THE APPLEII Review, Fall/Winter 1986, p. 12.
- <5> Golding, Val J. "Applesoft From Bottom To Top", CALL-A.P.P.L.E. IN DEPTH #1, 1981, p. 8.
- <6> Bernstein, Jeff. GENie, A2 ROUNDTABLE, Apr 1991, Category 2, Topic 16.
- <7> Arkley, John. (personal telephone call), Sep 9, 1991.
- <8> ----- "Apple Patches", PEEKING AT CALL-A.P.P.L.E., VOL 1, 1978, p. 40.
- <9> Weyhrich, Steven. "Applesoft Miscellaneous Information", APPLESOFT CONCORDANCE V1.0, Dec 1989.
- <10> Kamins, Scott. "Appendix D Reserved Words", APPLESOFT BASIC PROGRAMMER'S REFERENCE MANUAL, 1982, 1983.
- <11> Aldrich, Darrell. "The Computer Faire And The Apple", PEEKING AT CALL-A.P.P.L.E., VOL 2, 1979, p. 158.

--- End of Document ---