Hi-Res Packing For the APPLE II

by **R. M. Mottola**
Cyborg Corp.
342 Western Ave.
Boston, MA 02135

It never fails. No matter what kind of program I write, if it makes use of high resolution graphics it's always too long. With HIRES screen #1 at $2000 and screen #2 at $4000, Applesoft Basic programs that make use of HIRES graphics must be either less than 6K in length or less than 14K in length. It must be some corollary of Murphy's Law that explains why the programs I write always come out just a few bytes too long. Ideally a 48K system with DOS should leave about 28K for your program and variables after 8K is allocated to the HIRES screen and 2K to system requirements. Why doesn't it work that way?

The main reason for this is that the memory allocated to the HIRES screen is located right smack in the middle of the area used to store Basic programs. Why they put them there is a mystery to me, although I might guess that when the hardware was designed (back in the olden days of expensive RAM and the idea that this little computer would be simply a "hobby" item) putting the screens on the 8K and 16K boundaries probably made a lot of sense.

In order to make full use of the available memory, it would be nice if we could somehow load our Applesoft Basic program "around" the HIRES screen we wish to use. The short machine language routine listed at the end of this article will do just that. Once BLOADED into page three of memory, you can modify any Applesoft program that's longer than 6K in length to include the HIRES screen #1 right in the program. The modified program will run, list, and otherwise behave perfectly normal. However, it will have a big hole in it, the HIRES screen, starting at $2000 and ending at $3FFF. As an added bonus, that hole will contain nothing but zeros. In effect, the HIRES screen will be pre-cleared so you will not have to explicitly clear it upon entering the HIRES mode the first time.

HOW TO LOAD AND USE IT

The easiest way to Type in the program is to put your Apple into the Monitor Mode with the asterisk (*) as the Prompt character. Then type the Machine Language directly into memory beginning at Address $300 Hex (See page 44 of the Apple Reference Manual for directions). Your first entries should look like this:

```
*300:18 A5 B0 69 etc.
```

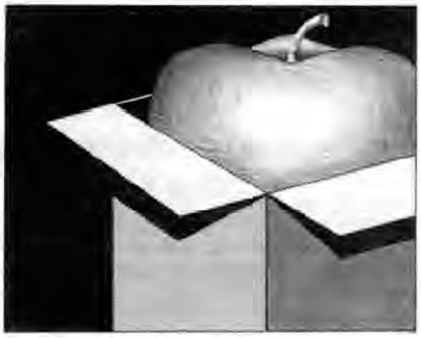
The Save on Disk using **BSAVE** (Program Name), **A$300,L$AF**

Use of the HIRES pack routine is very simple. First, BLOAD the routine at $300. Then, LOAD into memory the Applesoft program you wish to modify. Once loaded, a CALL 768 will do the job of embedding the HIRES screen into your program. The whole process takes about a second. After that, SAVE your newly packed program onto disk. It's a very good idea to save it under a different name from the unmodified version of the same program. The reason for this will be explained in the section called "RESTRICTIONS".

HOW IT WORKS

In order to understand just how this utility works, it's essential to understand how an Applesoft program is stored in memory. The following discussion will attempt to describe this.

An Applesoft program is stored in memory as a singly linked list. Each line of the program contains in it an absolute pointer to the next line. These pointers are the first thing on each line. Therefore, the pointer on any line will point to the pointer of the next line, which points to the pointer of the next line. . . on and on to the end of the program. A block diagram of an Applesoft program line will look like this:



2 BYTE ABSOLUTE POINTER TO NEXT LINE	2 BYTE LINE NUMBER	TOKENIZED BASIC PROGRAM	END OF LINE CHARACTER (0)

Since each line contains only one pointer which points to the next line in sequence, it can be seen that the only way to find something in a program is to start at the beginning and follow the chain of pointers through the program until you find what it is you are looking for, or until you run out of program — whichever comes first.

In our little routine, what we are looking for is the first line that lives at an address greater than $1FFF or, in other words, is located in the same memory locations as the beginning of HIRES page #1. The routine just goes down the chain of pointers, looking for one that points past $2000. Once found, it takes all of the rest of the program and moves it up to location $4000, just past the end of the HIRES screen. Next it clears (zeros) all of the memory locations from $2000 to $3FFF. Finally, it changes the pointer of the line that pointed to the first line that was moved to $4000, so that it does indeed point to the new position of that line. Now the list is again linked properly and will behave normally as a Basic program.

SOME RESTRICTIONS

The program listed will work just fine on an Apple II Plus or any Apple using Applesoft in ROM. It will not work for either Integer Basic programs or the RAM (cassette tape) version of Applesoft.

Once a program has been modified by this routine, almost any changes that you make to that program (DELetions, new lines, changed lines, etc.) will affect the positioning of the embedded HIRES screen. Although the program will look fine, as soon as it is run it will probably over-write portions of itself. You should never make changes to a program that has been modified by this utility. That's why it's a good idea to always keep two copies of the program on disk — one normal, the other modified with the embedded HIRES screen. The normal copy should not be run. Use it only as your source. That way, if your program does require some changes, you can make those changes to the normal version. Then, load and modify the newly changed version and save the it under a different file name. This will assure that you always have the ability to make program changes when required.

Remember that this routine will only be useful for programs longer than 6K in length. In fact, if you attempt to use it on a program that's less than 6K in length, crazy things will happen to your system.

The routine uses HIMEM to determine if there is enough room in memory to include your program with the embedded HIRES screen. It will exit with an error if there isn't enough room. Therefore, it is important that you set HIMEM to the same value that it will be set to when the program is to be run. Do this before you attempt to modify any program with this utility. Remember, the modified program will be at least 8K longer than the unmodified version, because it will have the HIRES screen packed into it.

Whenever DOS does a LOAD, it does a little more than just load the program into memory from the disk. One of the extra things it does is re-link the list that the Basic code is stored as. Usually this has no consequence, but in this case it has the unfortunate result of un-linking all of the code on the high side of the HIRES screen. To prevent this, a special technique must be used to load in a program. First, BLOAD into page three (A$300) the short routine named "LOAD PATCH" provided. A statement of the form:

```
100 PRINT D$; "BLOAD LOAD PATCH, A$300"
```

will do. Then perform the following POKES, to direct the DOS LOAD routine to the patch:

```
110 POKE 40288,0 : POKE 40289,3
```

The addresses provided in the above POKES are for a 48K system. If your system has less memory you must adjust them accordingly.

Once the POKEs have been done, you are ready to LOAD and/or RUN your modified program. After the program is loaded into memory it's a good idea to reset the locations POKEd into above. You should have a line like:

```
10 POKE 40288,242 : POKE 40289,212
```

early in your program. This will return DOS to it's normal operation.

Next time you're writing a lengthy program that makes use of the Apple's HIRES graphics, don't worry if you're fast approaching the end-of-avilable-memory wall. Use this little utility and gain a lot of extra space for your program and variables.

```
0800            ;***********************
0800            ;*                     *
0800            ;* APPLE II MACHINE    *
0800            ;* LANGUAGE UTILITIES  *
0800            ;*                     *
0800            ;* CYBORG CORPORATION  *
0800            ;* BOSTON, MASS.       *
0800            ;*                     *
0800            ;* R. M.  MOTTOLA      *
0800            ;*                     *
0800            ;***********************

0800            ;***********************
0800            ;*                     *
0800            ;* HIRES PACK ROUTINE  *
0800            ;*                     *
0800            ;***********************

0800            PT1L    EPZ $18
0800            PT1H    EPZ $19
0800            PT2L    EPZ $1A
0800            PT2H    EPZ $1B
0800            PT3L    EPZ $1C
0800            PT3H    EPZ $1D
0800            A1L     EPZ $3C
0800            A1H     EPZ $3D
0800            A2L     EPZ $3E
0800            A2H     EPZ $3F
0800            A4L     EPZ $42
0800            A4H     EPZ $43
0800            PROGL   EPZ $67
0800            PROGH   EPZ $68
0800            HIMEML  EPZ $73
0800            HIMEMH  EPZ $74
0800            ENDL    EPZ $AF
0800            ENDH    EPZ $B0
0800            NXTA4   EQU $FCB4
0800            MOVE    EQU $FE2C
0800            PRERR   EQU $FF2D
0300                    ORG $300
0300                    OBJ $800
0300            ;SEPARATES ANY APPLESOFT PROGRAM
0300            ;AND LEAVES SPACE IN THE MIDDLE
0300            ;OF IT FOR HIRES PAGE #1.
0300 18         SEPAR   CLC
0301 A5B0               LDA ENDH
0303 6920               ADC #$20
0305 C574               CMP HIMEMH
0307 9003               BCC SEP1
0309 4C2DFF             JMP PRERR
030C A567       SEP1    LDA PROGL
030E 851C               STA PT3L
0310 A568               LDA PROGH
0312 851D               STA PT3H
0314 A51A       SEP2    LDA PT2L
0316 8518               STA PT1L
0318 A51B               LDA PT2H
031A 8519               STA PT1H
031C A51C               LDA PT3L
031E 851A               STA PT2L
0320 A51D               LDA PT3H
0322 851B               STA PT2H
0324 A000               LDY #$0
0326 B11C               LDA (PT3L),Y
0328 48                 PHA
0329 C8                 INY
032A B11C               LDA (PT3L),Y
032C 851D               STA PT3H
032E 68                 PLA
032F 851C               STA PT3L
0331 A51D               LDA PT3H
0333 C920               CMP #$20
0335 90DD               BCC SEP2
0337 88                 DEY
0338 38                 SEC
0339 A900               LDA #$0
033B E51A               SBC PT2L
033D 851C               STA PT3L
033F A940               LDA #$40
0341 E51B               SBC PT2H
0343 851D               STA PT3H
0345 A900               LDA #$0
0347 9118               STA (PT1L),Y
0349 8542               STA A4L
034B C8                 INY
034C A940               LDA #$40
034E 9118               STA (PT1L),Y
0350 8543               STA A4H
0352 88                 DEY
0353 A51A               LDA PT2L
0355 853C               STA A1L
0357 A51B               LDA PT2H
0359 853D               STA A1H
035B A5AF               LDA ENDL
035D 853E               STA A2L
035F A5B0               LDA ENDH
0361 853F               STA A2H
0363 A000               LDY #$0
0365 202CFE             JSR MOVE
0368 98                 TYA
0369 853C               STA A1L
036B A920               LDA #$20
036D 853D               STA A1H
036F A9FF               LDA #$FF
0371 853E               STA A2L
0373 A93F               LDA #$3F
0375 853F               STA A2H
0377 98         ERASE   TYA
0378 913C               STA (A1L),Y
037A 20B4FC             JSR NXTA4
037D 90F8               BCC ERASE
037F 18                 CLC
0380 A51C               LDA PT3L
0382 65AF               ADC ENDL
0384 85AF               STA ENDL
0386 A51D               LDA PT3H
0388 65B0               ADC ENDH
038A 85B0               STA ENDH
038C A900               LDA #$0
038E 851A               STA PT2L
0390 A940               LDA #$40
0392 851B               STA PT2H
0394 A000       SEP3    LDY #$0
0396 18                 CLC
0397 B11A               LDA (PT2L),Y
0399 651C               ADC PT3L
039B 911A               STA (PT2L),Y
039D 48                 PHA
039E C8                 INY
039F B11A               LDA (PT2L),Y
03A1 651D               ADC PT3H
03A3 911A               STA (PT2L),Y
03A5 851B               STA PT2H
03A7 68                 PLA
03A8 851A               STA PT2L
03AA B11A               LDA (PT2L),Y
03AC D0E6               BNE SEP3
03AE 60                 RTS
                END     END
```

Hi-Res Packing (NIBBLE #4)

R.M. Mottola points out that the LOAD PATCH referred to in NIBBLE #4 (page 41) is not contained in the article. The code for this routine is listed below!

```
300: 20 65 D6 18 A0 01 A5 69
308: 85 AF A5 6A 85 B0 4C 3C
310: D4
```

This code enters Basic without re-linking the Basic code. It allows programs with embedded Hi Res Screens to run normally. It is called from DOS 'LOAD' and is normally found at $D4F2 in the Applesoft ROM.

R.M. also has indicated that the Hi Res Pack Routine has a limitation and will not work for programs longer than 12K in length. A memory list for a new routine (which removes this limitation) follows:

```
9513: A9 13 85 73 A9
9518: 95 85 74 18 A5 B0 69 20
9520: C5 74 90 03 4C 2D FF A5
9528: 67 85 1C A5 68 85 1D A5
9530: 1A 85 18 A5 1B 85 19 A5
9538: 1C 85 1A A5 1D 85 1B A0
9540: 00 B1 1C 48 C8 B1 1C 85
9548: 1D 68 85 1C A5 1D C9 20
9550: 90 DD 88 38 A9 00 E5 1A
9558: 85 1C A9 40 E5 1B 85 1D
9560: A9 00 91 18 85 42 C8 A9
9568: 40 91 18 85 43 88 A5 1A
9570: 85 3C A5 1B 85 3D A5 AF
9578: 85 3E A5 B0 85 3F A0 00
9580: 38 A5 3E E5 3C 48 A5 3F
9588: E5 3D 18 65 43 85 43 68
9590: 85 42 B1 3E 91 42 38 A5
9598: 42 E9 01 85 42 B0 02 C6
95A0: 43 38 A5 3E E9 01 85 3E
95A8: B0 02 C6 3F A5 3F C5 3D
95B0: B0 E0 A5 3E C5 3C B0 DA
95B8: 98 85 3C A9 20 85 3D A9
95C0: FF 85 3E A9 3F 85 3F 98
95C8: 91 3C 20 B4 FC 90 F8 18
95D0: A5 1C 65 AF 85 AF A5 1D
95D8: 65 B0 85 B0 A9 00 85 1A
95E0: A9 40 85 1B A0 00 18 B1
95E8: 1A 65 1C 91 1A 48 C8 B1
95F0: 1A 65 1D 91 1A 85 1B 68
95F8: 85 1A B1 1A D0 E6 60
```