



**UNLOCKED and  
UNPROTECTED**

Compatible with

- Apple IIe
- Apple II+
- Apple II

**Beagle Bros**  
Micro Software Inc.

# Alpha Plot

**Hi-Res Apple Graphics/Text Utility**

**by Bert Kersey & Jack Cassidy**

## HI-RES DRAWING

■ Create attractive pictures and charts on both hi-res pages, all **appendable to your Applesoft programs.**

■ Keyboard or paddle control; three different cursor styles available.

■ Optional Rubber Band (Xdraw) Cursor— see lines before you draw!

■ Plot in any color, color mix or REVERSE (background opposite).

■ One keystroke plots lines, circles, boxes and ellipses, filled or not.

## HI-RES TEXT

■ Attractive upper & lower case type in any Apple hi-res color or reverse.

■ Attractive professional-looking **proportionally-spaced type!**

■ Easy-to-read lower case with true descenders. No hardware required.

■ Adjustable typing size with matching-height flashing hi-res cursor.

■ Adjustable leading and kerning (space between letters and text lines) for optimum text readability.

■ Multi-directional typing— sideways for graphs, and even upside down!

■ Alpha Plot type is positionable anywhere on either hi-res page with no vtab or htab restrictions.



## BONUS HI-RES UTILITIES

■ **SCRUNCH:** Store hi-res images in as little as **1/3 normal disk space.** Prevent DISK FULL errors! Compressed pictures load much faster too.

■ **SHIFTER:** Relocate any section of an image to any screen location on either hi-res page.

■ **SUPERIMPOSE:** Blend any two hi-res images in one of four ways.

■ **IMAGE INVERSER:** Instantly convert a hi-res image to complementary colors.

■ Convert hi-res pictures to LO-RES and back! Create fascinating abstract images of your completed drawings.

## PLUS—

### APPLE TIP BOOK #4

Featuring informative articles and program listings to teach you more about programming your Apple and to help you make the most of it's amazing graphics capabilities!

### Free PEEKS & POKES

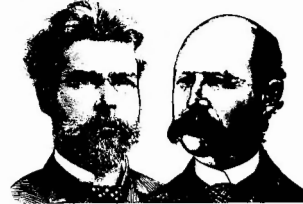
#### 11x17 Wall Chart enclosed

Apple's PEEKS, POKES, POINTERS and CALLS on one heavy-duty poster. An indispensable programming tool!

Beagle Bros presents

# Apple Tip Book #4

A New Assortment of  
APPLE II TIPS & TRICKS



Plus Complete Instructions for Using

## Alpha Plot

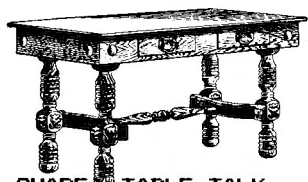
Hi-Res Text/Graphics Utility  
by Bert Kersey & Jack Cassidy

Entire contents Copyright © 1982, Beagle Bros. Micro Software  
4315 Sierra Vista, San Diego, CA 92103, (714) 296-6400.

"Apple", "Apple II", and the Apple Logo are all  
registered trade marks of the Apple Computer Company.

### TABLE OF CONTENTS

<b>TIP BOOK #4. . . . .</b>	<b>2</b>	<9> Cursor Speed. . . . .	33
-----		<0> Ellipses. . . . .	33
<b>ALPHA PLOT</b>		<: > Boxes . . . . .	33
<b>INSTRUCTIONS. . . . .</b>	<b>26</b>	<-> Lines . . . . .	33
General Information. . . . .	26	<Reset> Quit. . . . .	34
Cursor Moves . . . . .	27	<b>TYPING MODE. . . . .</b>	<b>34</b>
<b>DRAWING MODE . . . . .</b>	<b>28</b>	Features. . . . .	34
Special Keys . . . . .	28	Ctrl-O Options. . . . .	35
<1> Color Selection . . . . .	29	<M> Mode. . . . .	35
<2> Notes On/Off. . . . .	29	<R> Rotate Cursor . . . . .	35
<3> Grid On/Off . . . . .	29	<S> Size of Type. . . . .	35
<4> Draw, Type,		<L> Leading . . . . .	36
Save, Load. . . . .	30	<K> Kerning . . . . .	36
<5> Clear Screen. . . . .	30	Possible Problems. . . . .	36
<6> Image Options . . . . .	31	<b>ALSO ON THE DISK . . . . .</b>	<b>37</b>
<7> Key/Paddle. . . . .	32	Hi-Lo Plot. . . . .	37
<8> Modify Cursor . . . . .	32	Scrunch . . . . .	38



THIS IS  
TIP BOOK  
#4.

### SHAPE TABLE TALK

Let's assume you know nothing about shape tables (you're not the only one). Here are a few little experiments that will at least give you a taste of the fantastic animation possibilities.

USING shape tables is easy; it's MAKING them that is usually difficult. If you've looked in the Applesoft Manual in the shape table chapter, you'll see why. I'm not going to go into how to make shape tables here. I have just written a disk called "Apple Mechanic" that will write shape tables FOR you; all you do is draw your shape on the screen (as in Alpha Plot) and the shape table is automatically written by your Apple. Many good features are included (including Apple Tip Book #5).

Here is a sample shape table containing five shapes. You will recognize a few of the shapes from Alpha Plot. Type FF (return) to clear memory. Now enter the monitor by typing:

**CALL -151 (return)**

You should now have an asterisk and a flashing cursor. Enter the following shape table--

```
6000- 05 00 10 00 30 00 40 00
6008- 50 00 58 00 FF FF FF FF
6010- 2D 2D 2D 3E 3F 3F 37 2D
6018- 2D 2D 3E 3F 3F 37 2D 2D
6020- 2D 3E 3F 3F 37 2D 2D 2D
6028- 3E 3F 3F 3F 00 FF FF FF
6030- 08 0C D6 AA DF 63 1C 08
6038- 1C 07 00 FF FF FF FF FF
6040- 12 3F 20 64 2D 15 36 1E
6048- 07 00 FF FF FF FF FF FF
6050- 07 00 FF FF FF FF FF FF
6058- 3F 37 2D 35 37 4D C0 2C
6060- 1C 3F 3F 4C E5 07 00 FF
6068- FF FF FF FF FF FF FF FF
```

#1 ■

#2 ✕

#3 ○

#4 .

#5 ➡

(FF's are  
wasted.)

In case you've never entered machine code before, just type:

**6000: 05 00 10 00 30 00 40 00 (return)**

and

**6008: 50 00 58 00 FF FF FF FF (return)**

and so on. Notice the COLON (not a HYPHEN) after the four digit number; very important! Now save the shape table by typing:

**BSAVE SHAPES,A#6000,L#70**

Any time you want these shapes back, type:

**BLOAD SHAPES**

Now tell your Apple where you put the shape table by typing:

**POKE 232,0: POKE 233,96**

You could make these pokes part of a program. 232-233 is the Start of Shape Table pointer. See your Peeks &

Fokes Chart.

Now we can play with the Apple commands DRAW, XDRAW, ROT and SCALE (I had an uncle once with rot and scale). Try each of the following programs.

```
=====
SHAPE DISPLAY
=====
```

```
15 TEXT : HGR
16 ROT= 0: SCALE= 1: HCOLOR= 3
20 FOR SH = 1 TO 5
30 DRAW SH AT SH * 21,150
40 NEXT SH
50 HOME : VTAB 21
60 HTAB 4: PRINT "1 2 3 4 5"
```

```
=====
ROT TEST
=====
```

```
30 HOME : HGR : SCALE= 1: HCOLOR= 3
35 HPLLOT 0,150 TO 279,150
40 FOR X = 50 TO 250 STEP 1
80 R = R + 16: IF R > 48 THEN R = 0: REM INCREASES ROT
   TATION 90 DEGREES
82 VTAB 21: HTAB X / 7 - 4: PRINT SPC( 1);"ROT:";R;
   : CALL - 868: REM PRINTS ROT VALUE
83 VTAB 22: HTAB X / 7 - 2: PRINT SPC( 1);"X:";X: REM
   PRINTS X POSITION
85 ROT= R: XDRAW 5 AT X,140: REM DRAWS ARROW
90 FOR I = 1 TO 99: NEXT I
95 XDRAW 5 AT X,140: REM ERASES ARROW
100 NEXT X
```

```
=====
SCALE TEST
=====
```

```
30 TEXT : HOME : HGR : ROT= 0
40 FOR S = 1 TO 10
80 SCALE= S
82 VTAB 21: HTAB 10: PRINT "SCALE:";S;: CALL - 958
85 XDRAW 1 AT 70,30: REM DRAWS BOX
90 IF S = 1 THEN FOR I = 1 TO 500: NEXT I
95 XDRAW 1 AT 70,30: REM ERASES BOX
100 NEXT S: GOTO 40
```

Fool around with these programs and change some of the numbers. Substitute DRAW for XDRAW and see what happens (be sure you have HCOLOR set for something other than black). XDRAW draws in the OPPOSITE of whatever the background color is. That's how you can erase a drawing; just XDRAW over it in exactly the same position! Notice how SCALE, when it's bigger than 1, blows up a solid shape in a linear fashion. Too bad it won't blow up solids.



### KABLOOEY!

Here's an experiment to demonstrate Hi-Res collisions, handy if you feel the urge to write a SDTFS\* game. I have read two articles, each about four blocks long about how to simulate Lo-Res's SCRNB function for determining the color of a Hi-Res point. Well, forget it. There's a handy peek you can do of location 234 that tells you if the last XDRAW was done over a black or non-black point. This program pretty much explains itself. Oh, BE SURE THE SHAPE TABLE FROM THE PREVIOUS ARTICLE IS IN MEMORY and the pointers are set (POKE 232,0: POKE 233,96), or this program won't work.

### =====

#### COLLISION TEST

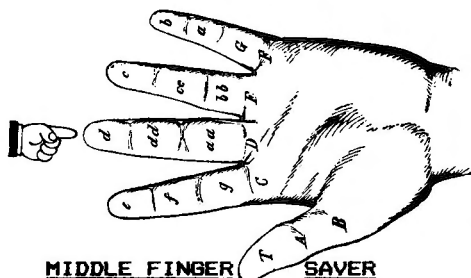
### =====

```
20 TEXT : HOME : HGR : ROT= 0: SCALE= 1:N = 100
30 HCOLOR= 6: HPL0T 250,0 TO 250,191: HPL0T 30,0 TO
   30,191:POKE 232,0: POKE 233,96
40 FOR X = N + 2 TO 279
50 XDRAW 3 AT X,100
60 XDRAW 3 AT X,100
70 IF PEEK (234) = 0 THEN NEXT X
80 FOR N = X - 1 TO 0 STEP - 1
90 XDRAW 2 AT N,100
100 XDRAW 2 AT N,100
110 IF PEEK (234) = 0 THEN NEXT
120 GOTO 40
```

### WARNING DEPARTMENT

Rumor has it that you should always OPEN YOUR DRIVE DOORS and/or REMOVE ALL DISKS from your drives BEFORE TURNING OFF YOUR APPLE. You risk disk damage if you don't. And while we're issuing warnings-- IF you leave your Apple on and unattended for a long time AND a temporary power failure occurs, the Autostart ROM will attempt to reboot. IF your drive door is open OR no disk is in your drive, your drive could run indefinitely (or until the next power failure, whichever comes first).

\*SDTFS=SHOOT DOWN THE FLYING SAUCER



### MIDDLE FINGER SAVER

Think of the time you'll save! In spite of what the DOS Manual says, NO COMMAS or spaces are necessary when you do a "MON I,C,D" or a "NOMON I,C,D". Another comma recommended by the Manual is the one after the CATALOG command and before the drive number. "CATALOGD1" gets the job done just fine!

### REM MAKER

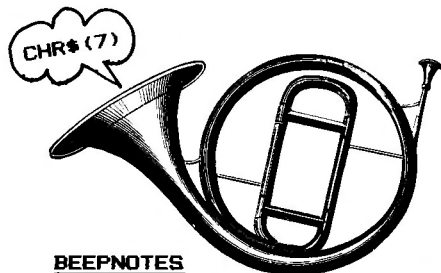
Here's the easiest way we know to make REM's show up: Type "100 REM (ctrl-J) (ctrl-J) THIS IS A REMARK (ctrl-J)". Now you have some air above and below your REM, making it easier to find in a listing.

### DON'T EVER DO THIS:

```
!CALL -151
```

```
*C055
```

Thank you.



### BEEPNOTES

First of all, to "print" a beep, you type "PRINT", then a quote mark, then a ctrl-G (audible, but invisible), and then another quote mark. Follow that with a semi-colon if you don't want the screen to scroll. You will hear the ctrl-G in the listing as well as in the program, unless...

To prevent your program's ctrl-G beeps silent in your listings, LET G\$ = CHR\$(7) (same as ctrl-G). Now when you want a beep, simply PRINT G\$.

And, speaking of beeps, you can use ctrl-G as a nifty de-bugging device-- Often the reason a certain program line doesn't work is that the program NEVER GOT TO that line. To see if it did, simply put a "PRINT CHR\$(7)" at the beginning of the statement. RUN the program; if it beeps, you know it encountered the statement in question.



```

20 HOME :C$ = " "
30 FOR I = 1 TO 23: VTAB I: HTAB 1: PRINT C$;: HTAB 40:
   PRINT C$;: NEXT I
50 FOR I = 1 TO 24 STEP 23: VTAB I: HTAB 1: FOR J = 1 TO
   39: PRINT C$;: NEXT J,I
60 POKE 2039, ASC (C$) + 128: REM (PUTS C$ AT VTAB24,HT
   AB40)
70 NORMAL : VTAB 10:A$ = "VISI-FRAME": GOSUB 120: PRINT
   : PRINT A$: "BY WILL MUFFIN": GOSUB 120
80 VTAB 20:A$ = "(PRESS ANY KEY: )": GOSUB 120:PRINT CHR$
   (8): CHR$ (8):: REM (2 BACK-SPACES)
90 GET C$: PRINT C$: REM (NEW FRAME CHARACTER)
100 IF C$ > " " THEN 30
110 FOR I = 1 TO 26: PRINT : NEXT : END
120 HTAB 20 - LEN (A$) / 2: PRINT A$: RETURN : REM
   (CENTERS TITLE)

```

### REM OVE

If you're removing REMs from your program to save memory, you might want to instead REPLACE each statement with a single colon. That way, if you have any GOTOs to your REM lines, things will still function normally.

### TEXT OR HI-RES?

If your Apple monitor is a color tv, you can tell if the screen text you are reading is hi-res or text by looking for little color glitches between vertical strokes on letters such as m's. If you see them, it's hi-res. By the way, you won't see small (normal size) color text on the Apple. There just aren't enough dots available to do the job.

### ?ERROR?

When you encounter an error in hi-res, you get a beep, but usually no visible error message. Just hit reset or type TEXT to reveal the message.

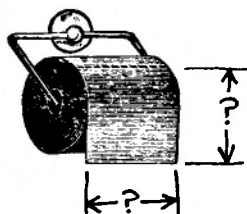
### 10 REM

#### ===== VOWEL SCRAMBLER =====

```

15 ERR$ = CHR$ (7) + "?SYNTAX ERROR":YES$ = CHR$ (7) +
   "!"CORRECT"
20 W$ = "":EFLAG = 0: HTAB 1: PRINT "J":
30 GET A$:W$ = W$ + A$: IF A$ < > CHR$ (13) THEN 70
35 IF LEN (W$) < 2 THEN 20
40 PRINT : PRINT : IF EFLAG THEN PRINT ERR$: GOTO 55
50 PRINT YES$:EFLAG = 0
55 FOR I = 1 TO 99: NEXT :W$ = LEFT$ (W$, LEN (W$) - 1)
   : IF NOT EFLAG AND W$ = "LIST" THEN LIST
60 GOTO 20
70 L = ASC (A$): IF A$ < > "A" AND A$ < > "E" AND A$ <
   > "I" AND A$ < > "O" AND A$ < > "U" THEN PRINT A
   $: GOTO 30
80 N = INT ( RND (1) * 5):X = 65 + N * 4 + (N = 3) * 2 +
   (N = 4) * 4: PRINT CHR$ (X): IF X < > L THEN EFLA
   G = 1
90 GOTO 30

```



### GUESSTIMATION

I have a theory: If you had a million people guess the height and width of a dollar bill, the AVERAGE guesses would be EXACTLY right! Well, test it for me; I only know four people, and one is my dog. Take any rectangular object, like a dollar bill or the dining room table, or the Apple, and have a dozen or so people guess its size. See how close their average guess comes to the actual size.

```

20 HOME : INVERSE : PRINT " SIZE GUESSER ": NORMAL : PRINT
30 PRINT "GUESS DIMENSIONS": INPUT "OF WHAT ITEM? ";ITEM
40 INPUT "NUMBER OF GUESSERS: ";N: DIM NAME$(N),W(N),H(N)
   : GOSUB 180
50 FOR I = 1 TO N: PRINT : PRINT "NAME OF GUESSER #";I;:
   INPUT "": NAME$(I)
60 PRINT "GUESS WIDTH OF ";ITEM$;: INPUT "": W(I)
70 PRINT "GUESS HEIGHT OF ";ITEM$;: INPUT "": H(I)
80 NEXT I: GOSUB 180
90 PRINT "ACTUAL WIDTH OF ";ITEM$;: INPUT "": W(0)
100 PRINT "ACTUAL HEIGHT OF ";ITEM$;: INPUT "": H(0)
110 GOSUB 180
120 HOME : PRINT "NAME","WIDTH";: HTAB 29: PRINT "HEIGHT"
   : GOSUB 180
130 FOR I = 1 TO N: VTAB 2 + I: HTAB 1: PRINT NAME$(I),W
   (I);: HTAB 29: PRINT H(I): NEXT
140 FOR I = 1 TO N:W = W + W(I):H = H + H(I): NEXT :WAVG
   = W / N:HAVG = H / N
150 GOSUB 180: PRINT "AVERAGE GUESS:",WAVG;: HTAB 29: PRINT
   HAVG
160 PRINT "ACTUAL SIZE:",W(0);: HTAB 29: PRINT H(0)
170 GOSUB 180: END
180 HTAB 1: FOR I = 1 TO 20: PRINT "- ";: NEXT : RETURN

```



### ONE-KEY COMMANDS!

Type this--

POKE 1013,76: POKE 1014,110: POKE 1015,165

Now, the command "&" (return) will CATALOG! Change the numbers after the commas to 76, 165 & 214 and "&" will LIST. 76, 18 & 217 will make "&" RUN. 76, 112 and 214 will make "&" CLEAR all variables to zero. When you type "&" (that's called an "ampersand"), the Apple jumps to location 1013 (\$3F5) to see what's happening there. The pokes discussed here tell the Apple what to do.





### POINTER CHECKER

This is really fun if you're into snooping around Apple's memory. When you run this program, it will print 88 addresses and their peeked values on the screen. What you do now is change something like a COLOR value or ROT, and run the program again. The most interesting changes take place on Page Zero (addresses 0-255), so let Line 30's START=0 or 100 or 200.

Example: In Line 30, LET START=0. Now RUN, and wait for the cursor to come back. Now, type "COLOR=0:RUN" (both commands on the same line to prevent scrolling). Most of the inverse arrows will disappear. The ones that DON'T DISAPPEAR are addresses whose values have CHANGED since the first time you ran the program. Now type "COLOR=2:RUN", and watch location 48. It should change from a 0 to a 34 (the current lo-res COLOR\*17). Two locations that almost ALWAYS change are 78 and 79, the random number field. Compare your results with your Peeks & Fokes Chart.

```
20 DIM A(23): FOR I = 0 TO 23: READ A:A(I) = A: NEXT
30 START = 0
40 IF START < 0 THEN START = START + 65536
50 VTAB 1: HTAB 1: FOR X = START TO 65535
70 PRINT " "; CHR$(8); SPC( (X < 10000) + (X < 1000) + (
  X < 100) + (X < 10)); X; P = PEEK(X): PRINT " ";
80 P1 = A(PEEK(37)) + PEEK(36): A1 = PEEK(P1): A2 = PEEK
  (P1 + 1): A3 = PEEK(P1 + 2)
100 PRINT P; IF P < 10 THEN PRINT CHR$(95);
110 IF P < 100 THEN PRINT CHR$(95);
120 B1 = PEEK(P1): B2 = PEEK(P1 + 1): B3 = PEEK(P1 + 2)
  IF A1 < > B1 OR A2 < > B2 OR A3 < > B3 THEN
  INVERSE: PRINT "<"; NORMAL: GOTO 150
130 PRINT SPC(1);
150 IF PEEK(37) < 22 THEN NEXT
170 VTAB 22: DATA 1024,1152,1280,1408,1536,1664,1792,1920
  ,1064,1192,1320,1448,1576,1704,1832,1960,1104,1232,13
  60,1488,1616,1744,1872,2000
```

### SLOT SEARCH

WE NEED YOUR HELP! Here's a program that reads your Apple's slots to see what's in them. We know that if a disk controller card is in a slot, the value for that slot is 162. The three printers we have checked out have returned a 24. A Novation Super-Cat Modem is a 255. Please let us know your results when you run this program with your equipment. Apparently, if a slot is empty, the value keeps changing, so look for a slot with an unchanging number.

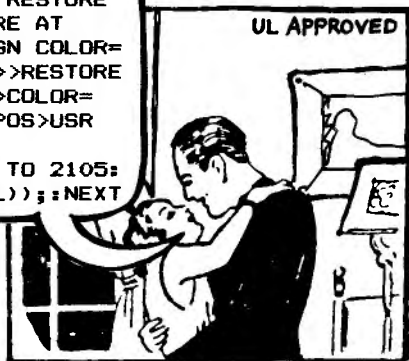
```
10 HOME : FOR X = 1 TO 20
20 FOR I = 1 TO 7: N = 49152 + 256 * I: PN = PEEK(N)
30 VTAB 4 + 2 * I: HTAB 1: PRINT "SLOT "; I; ": PEEK("; N;
  ") = "; PN; SPC(2)
40 VTAB 4 + 2 * I: HTAB 26
50 B(I) = A(I): A(I) = PN: IF A(I) < > B(I) THEN CALL -
  868: PRINT "(EMPTY)": GOTO 500
60 INVERSE
100 IF PN = 162 THEN PRINT "DISK DRIVE"
110 IF PN = 24 THEN PRINT "PRINTER"
500 NORMAL : NEXT I, X: VTAB 22
```

```

1 GOTO 2 AND POS COLOR=OR
  TO AND AT COLOR=-INT
  COLOR=INT USR AT RESTORE
  PLOT-COLOR= ^ -FRE AT
  COLOR=OR AT TO SGN COLOR=
  ABS+AT COLOR=SGR>>RESTORE
  PLOT PLOT SCRN(+>COLOR=
  TO SGN AT COLOR=POS>USR
  NEW PLOT
2 HOME: FOR L=2055 TO 2105:
  PRINT CHR$(PEEK(L));:NEXT

```

UL APPROVED



## PERPETUAL A<sub>1</sub>Contest!

Uncle Louie's Apple is an old one with only 0.6K, so he really has a problem with some of the software on the market today. He was pretty excited about his upgrade last Christmas-- we got him a 40-column board and a paddle. But he's still lacking in memory (in more ways than one), and needs some Two-Liners to play with.



**GIVE UNCLE LOUIE  
A HAND!**

### RULES:

Your program must be written in Applesoft or Integer BASIC and be no longer than two program lines, typable without a ?TOO LONG ERROR message. Decision of Uncle Louie is final. All programs will be judged on...

(a) how impressive the program is when run.

### PRIZES:

1st Prize: Any Beagle Bros disk  
 2nd Prize: Any Beagle Bros disk  
 3rd Prize: Any Beagle Bros disk  
 4th-9th Prizes: Haven't decided yet  
 10th Prize: Uncle Louie

The best 2-Liners will be printed in the next Tip Book and possibly make our TipDisk. Lookit-- It's really a GOOD way to get some free software. Send your Two-Liners in today!

CHECK OUT SOME OF OUR  
WINNERS -- NEXT PAGE.





Here are some  
of the winning  
entries in  
Uncle Louie's  
semi-perpetual  
Two-Liner contest.  
Give them a try!

=====  
APPLESOFT  
2-LINERS  
=====

```
0 REM DAVID LINKER-- SEATTLE, WA
1 GR : HOME : COLOR= 15:T = 39: FOR I = 0 TO 1: VLIN
  O,T AT I * T: HLIN O,T AT I * T: NEXT X = 5:Y =
  5:S = 0:T = 38: COLOR= 1: PLOT X,Y: FOR C = 1 TO
  14:A = RND (1) * T + 1:B = RND (1) * T + 1: COLOR=
  C + 1: PLOT A,B: COLOR= C: VTAB 23: PRINT "YOUR
  SCORE=";S: FOR I = 1 TO 50: IF (N = 15) OR (N <
  = C AND N > 0) THEN END
2 N = PEEK ( - 16368):N = PEEK ( - 16384):G = (N =
  21) - (N = 8):D = (N = 90) - (N = 65): ON ABS (
  G + D) + 1 GOTO 2:X = X + G:Y = Y + D:N = SCRNI
  X,Y):S = S + 200 * (N = C + 1):Z = PEEK ( - 163
  36): PLOT X,Y: NEXT :S = S + 100: COLOR= 0: PLOT
  A,B: NEXT : VTAB 23: HTAB 12: PRINT S;" YOU WIN!
  ": FOR I = 1 TO 9: CALL - 1052: NEXT
3 REM USING THE A, Z & ARROW KEYS, TRY TO GET THE DOT
  WITHOUT RUNNING INTO YOUR OWN TRAIL.
```

---

```
0 REM DAVID GERSHON-- SEAL BEACH, CA
1 HGR : HOME : FOR C = 1 TO 7:RT = 15: HCOLOR= C: FOR
  R = 5 TO 150 STEP 10: HPLLOT O,R TO RT,155 TO 270
  ,155 - R TO 270 - RT,0 TO O,R:RT = RT + 15: NEXT
2 FOR R = 10 TO 60 STEP 10: HPLLOT 140,10 + R TO 140 +
  R,75 TO 140,150 - R TO 140 - R,75 TO 140,10 + R:
  NEXT R: NEXT C: GOTO 1
```

---

```
0 REM MARK BACHMAN-- AUSTIN, TX
1 TEXT : HOME : PRINT TAB( 13);"KEYBOARD ORGAN": FOR
  X = 24576 TO 24591: READ Y: POKE X,Y: NEXT X: DATA
  172,0,192,173,48,192,234,234,234,234,136,208,24
  9,76,0,96
2 CALL 24576
```

---

```
0 REM I.O.SOCKET-- SAN DIEGO, CA
1 NORMAL : VTAB 1: HTAB 1: INPUT "WHAT'S YOUR NAME?
  ";N$: SPEED= 234:SP$ = "" : ONERR GOTO 1
2 SP$ = SP$ + "." : INVERSE : PRINT " ";N$;" " : NORMAL
  : PRINT SP$: GOTO 2
```

```
=====
INTEGER BASIC
2-LINERS
=====
```

```
0 REM DAVE MADDEN-- LISLE, IL
1 CALL -936: TAB 10: PRINT "U.F.O. TAKEOFF EFFECT"
: TAB 13: PRINT " BY DAVE MADDEN"
2 POKE 766,1: POKE 765,16: FOR A=1 TO 255: FOR B=
1 TO 75: FOR I=B TO 1 STEP -A: POKE 767,I: CALL
-10473: NEXT I,B,A: GOTO 2
3 REM REQUIRES PROGRAMMER'S AID CHIP
```

---

```
0 REM DAVE MADDEN AGAIN
1 CALL -936: TAB 11: PRINT " HYPERDRIVE EFFECT": TAB
13: PRINT "BY DAVE MADDEN":A=100
2 POKE 766,1: POKE 765,2: FOR I=1 TO A: POKE 767,
I: CALL -10473: NEXT I:A=A-1: IF A<=-30 THEN A=
100: GOTO 2
3 REM REQUIRES PROGRAMMER'S AID CHIP
```

---

```
0 REM DARRYL FIELDS-- PHOENIX, AZ
1 FOR P=0 TO 3000: NEXT P: GR : PRINT : PRINT : COLOR=
RND (16): FOR Q=0 TO 39: HLIN 0,39 AT Q: NEXT
Q:N=0:D= RND (25)
2 COLOR= RND (15):B= RND (15)+24:A= RND (B):Y= RND
(15)+24:X= RND (Y): FOR Z=A TO B STEP 2: HLIN X,
Y AT Z: NEXT Z:N=N+1: IF N<D THEN 2: GOTO 1
```

```
=====
APPLESOFT AND INTEGER
2-LINER
=====
```

```
0 REM MARC RINGUETTE-- TERRACE, B.C., CANADA
1 PRINT "OPENX": PRINT "WRITEX": PRINT "CALL-151": PRINT
"2:AD 30 C0 88 D0 04 C6 01 F0 08 CA D0 F6 A6 00
4C 02 00 60 40 00": PRINT "INT": PRINT "9POKE0,A
:POKE1,B*42:CALL2:RETURN": PRINT "3A=66:B=4:GOSU
B9:A=56:B=3:GOSUB9:B=1:GOSUB9:A=83:B=6:GOSUB9:A=
74:B=2:GOSUB9:A=66:GOSUB9"
2 PRINT "5A=63:GOSUB9:A=56:GOSUB9:A=49:GOSUB9:A=74:B
=6:GOSUB9:FORJ=1TO60:NEXTJ:A=66:B=4:GOSUB9:A=59:
B=3:GOSUB9:B=1:GOSUB9": PRINT "7A=56:B=6:GOSUB9:
A=49:B=2:GOSUB9:A=43:GOSUB9:GOSUB9:A=56:B=6:GOSU
B9:END": PRINT "RUN": PRINT "CLOSEX": PRINT "EXE
CX"
3 REM TO ENTER THIS PROGRAM IN ONLY 2 LINES, YOU MU
ST TYPE IT WITH NO SPACES AND SUBSTITUTE A "?"
FOR EACH "PRINT". ALSO, ADD A CTRL-D JUST AFTER
THE FIRST QUOTE MARK ON "OPENX", "WRITEX" &
"CLOSEX".
4 REM YOU MUST HAVE BOTH APPLESOFT AND INTEGER IN YOUR
APPLE TO RUN THIS PROGRAM. IMPORTANT: SAVE
BEFORE YOU RUN!
```



10 REM

# TREASURE HUNT (requires paddles)

Don't get all excited; this "game" will NEVER make the Top Thirty, but it WILL teach you a few things you may not already know. Line 15 sets up a shape table of a single dot (the number 5 at 24580) that can be stretched by the SCALE command. The rems pretty much explain the rest.

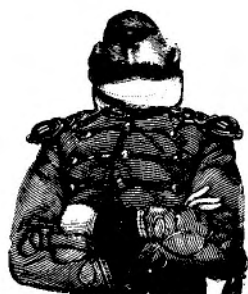
```

15 POKE 24576,1: POKE 24577,0: POKE 24578,4: POKE 24579,
   0: POKE 24580,5: POKE 24581,0: REM SHAPE TABLE FOR
   LINE
20 POKE 232,0: POKE 233,96: REM POINTERS TO START OF SHAPE
   TABLE
30 PX = 279 / 255: PY = 191 / 255: REM PADDLE LIMITS

40 HGR : REM CLEAR & SET HI-RES SCREEN
50 HOME : VTAB 21: HTAB 4: REM CLEAR TEXT & PUT CURSOR
   IN VIEW
60 PRINT "CAN YOU FIND THE HIDDEN TREASURE?": VTAB 24: HTAB
   16: PRINT "(HINT: USE THE PADDLES.)": REM SEMI-COLON
   PREVENTS VTAB 24 SCROLL-UP
70 XXX = 100: YYY = 50: HCOLOR= 3: HPLLOT XXX - 10, YYY - 10
   TO XXX + 10, YYY + 10: HPLLOT XXX - 10, YYY + 10 TO XX
   X + 10, YYY - 10: REM DRAW AN X
80 X = PX * PDL (0): Y = PY * PDL (1): REM READ PADDLES
90 SCALE= 255: ROT= 0: XDRAW 1 AT 0 + 24 * (X > 140), Y: REM
   DRAW HORIZONTAL LINE
100 SCALE= 191: ROT= 16: XDRAW 1 AT X, 0: REM DRAW VERTICAL
   LINE
110 IF (X < XXX + 2 AND X > XXX - 2) AND (Y < YYY + 2 AND
   Y > YYY - 2) THEN VTAB 22: HTAB 13: INVERSE: PRINT
   " YOU FOUND IT! ": FOR I = 1 TO 500: NEXT : NORMAL :
   VTAB 22: HTAB 13: PRINT SPC( 15)
120 VTAB 24: HTAB 1: PRINT "X=": INT (X): " Y=": INT (Y)
   : SPC( 3): REM SPC(3) CLEARS EXTRA CHARACTERS
130 SCALE= 255: ROT= 0: XDRAW 1 AT 0 + 24 * (X > 140), Y:
   SCALE= 191: ROT= 16: XDRAW 1 AT X, 0: REM ERASE BOTH
   LINES
140 IF PEEK ( - 16384) > 127 THEN 160: REM CHECK FOR KEY
   PRESS
150 GOTO 80: REM GO BACK
160 POKE - 16368,0: REM PREVENT KEYPRESS FROM PRINTING
   ON SCREEN
170 HOME : VTAB 22: PRINT "SO LONG!"
  
```

IF I WERE YOU,  
I'D PHONE IN SICK  
TOMORROW, AND  
PLAY WITH MY  
COMPUTER.





### WEIRDNESS DEPARTMENT

#### 1. Try this--

**BSAVE LONG PROGRAM,A\*9000,L\*7FFF**

This bsave will zap your text screen into lo-res and make little "vibration" marks on some of the lo-res color plots and even click the speaker (sometimes?). A Hopefully-Not-Too-Late Warning: Don't hit reset during disk access; you could foul up your disk. A good rule is to never hit reset while the "In Use" light is on. Open the drive door (gasp!) first if you must.

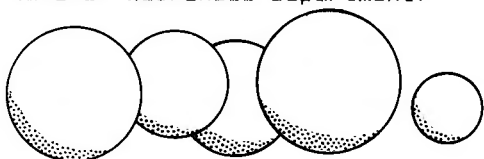
2. Long Applesoft token words like "INVERSE" and "NORMAL" fail to produce a carriage return when they appear near the right margin of listings; a real pain when you're printing listings.

3. If you DRAW or XDRAW without an HGR, HGR2 or similar command first, you will produce screen garbage.

4. RESET resets almost everything, but it won't reset SPEED to its normal 255 value.

5. Every OTHER time you CALL 64871 you get a ?SYNTAX ERROR.

(end of Weirdness Department)



### HI-RES CIRCLES

One of the most frequently-asked questions around here (next to "Who zapped my disk?") is "How do you draw a hi-res circle?" Well, here's one way--

```

20 HGR : HCOLOR= 3:ST = .1
30 POKE - 16301,0: HOME : VTAB 22: HTAB 1: INPUT "ENTER
  X CENTER, Y CENTER, RADIUS:";XCTR,YCTR,RAD:FLAG = 0
  :ST = .1: POKE - 16302,0
40 FOR I = 0 TO 6.3 STEP ST
50 X = RAD * COS (I) + XCTR
60 Y = (RAD / 4) * SIN (I) / .3 + YCTR
70 IF X > 279 OR X < 0 OR Y > 191 OR Y < 0 THEN FLAG = 0
  : GOTO 100
80 IF NOT FLAG THEN HPOINT X,Y:FLAG = 1
90 HPOINT TO X,Y
100 NEXT I : GOTO 30

```

Note the variable ST in Line 30. It determines how many "sides" the circle will have. Change it and see.



CHR\$(77); CHR\$(69);  
CHR\$(79); CHR\$(87);  
CHR\$(33); CHR\$(33)

### STRIPE-CAT

Another in a long series of good-for-nothing (but worthless) Beagle Bros programs. What this one does is write a text file called "C". To use it, type EXEC C. What you will get is an entirely different type of catalog. Oh, go AHEAD. Type it in.

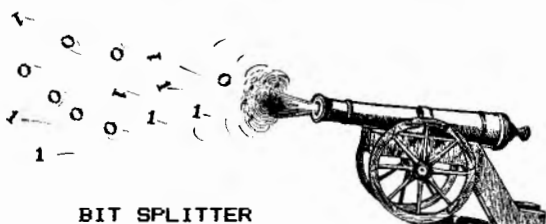
```
40 PRINT CHR$(4);"OPEN C"
50 PRINT CHR$(4);"WRITE C"
60 PRINT "POKE 44452,12"
70 PRINT "POKE 44605,11"
80 PRINT "POKE 33,37"
90 PRINT "POKE 50,63"
100 PRINT "CATALOG"
110 PRINT "TEXT:VTAB22:NORMAL:POKE44452,24:POKE44605,23"
120 PRINT CHR$(4);"CLOSE"
```

10 REM

=====

DOWNSCROLL

```
20 GOSUB 30: FOR N = A TO B STEP - C: FOR I = N TO D STEP
  - E: FOR J = I TO I + F: POKE J, PEEK (J - E): NEXT
  : NEXT : FOR I = N - G TO N - H: POKE I, PEEK (I + K)
  : NEXT : NEXT : VTAB 1: HTAB 1: CALL - 868: GOTO 2
  0
30 A = 2000:B = 1920:C = 40:D = 1132:E = 128:F = 39:G = 8
  96:H = 857:K = 856: RETURN
```



### BIT SPLITTER

If you are having a tough time figuring out binary numbers, here's a little demo that will help.

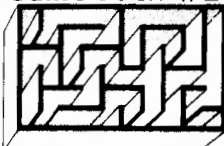
```
20 HOME : VTAB 8: HTAB 9
30 PRINT "128 64 32 16 8 4 2 1": HTAB 9
40 PRINT "---- -- -- -- -- -- --"
50 FOR J = 0 TO 255
60 FOR I = 1 TO 8:BIT(I) = INT (J / 2 ^ (I - 1)):BIT(I)
  = BIT(I) - 2 * INT (BIT(I) / 2): NEXT
70 VTAB 10: HTAB 4: PRINT SPC( J < 100): SPC( J < 10):
  INVERSE : PRINT J:; NORMAL : PRINT " : ";
80 FOR I = 8 TO 1 STEP - 1:B = BIT(I)
90 NORMAL : IF B = 1 THEN INVERSE
100 PRINT B:; NORMAL : PRINT SPC( 2):; NEXT
110 PRINT : NEXT
```

## Game Pack #1\*



- 1. TextTrain:** Race the on-screen clock with your text-format video "freight train." Real-time track switching & coupling simulations; hours of fun!
- 2. Sub Search:** Find & capture the invisible enemy subs on your Apple color graphics scope! Sound-enhanced scanner, tracer & instrument panel!
- 3. Pick-a-Pair:** A colorful Apple party game for all ages and skill levels! Uncover and remember the hidden graphics symbols to score big & win!

## Game Pack #2\*



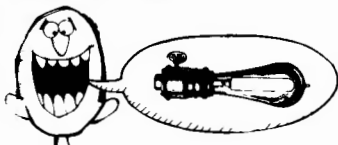
- 1. Wowzo:** Our challenging changeable maze game! Capture targets in a flexible maze, and outmaneuver your opponent before time runs out!
- 2. Elevators:** Keyboard control 4 elevators at one time in your CRT skyscraper. You'll need a computer to solve this one!
- 3. Quick-Draw!** You command two colorful gunmen who shoot it out on your Apple screen!

## Game Pack #3\*



- 1. Magic Pack:** Four mind-bending tricks in one fantastic Magic Show! Only you and your Apple know how to perform these amazing feats!
- 2. Slippery Digits:** A challenging & colorful number-action game for all ages. A great demonstration of your Apple's capabilities!
- 3. Oink!** A nerve-racking sound-enhanced video dice game with unpredictable results and lots of laughs!

## Game Pack #4\*



- 1. Buzzword:** A comical story-creator with endless possibilities. 5 changeable stories in memory plus a fascinating "Create Your Own Story" program!
- 2. Triple Digits:** A thinker's game with numbers. Score in four ways and outfox your opponent!
- 3. Corn Game:** A kids' guessing game involving 3 farm animals and endless supply of corn!

# dos boss™

## DISK COMMAND EDITOR

by Bert Kersey and Jack Cassidy

**Dos Boss** is an extremely versatile, easy-to-use Apple utility package that will customize your disk system and personalize your personal computer! Here are just SOME of Dos Boss's useful features—

**Rename DOS Commands** by simply entering the command you want changed (say "CATALOG") and your new command (say "CAT"). Now "CAT" will catalog your disks. Other changes are just as easy...

**Change the "Disk Volume"** heading to anything you want; your name, disk title or code; with or without the Volume Number. Inverse, Flash or Normal!

**"Save-Protect" your programs!** An unauthorized copy attempt will produce a "NOT COPYABLE!" message.

**One-key program selection!** Run programs by pressing only the key indicated on the screen. Instant free-space on disk with one key too!

**Customized Catalogs!** Create multi-columned catalogs that fit more file names on the screen. Catalog only the file-types you want (A, I, B and/or T). Omit or alter sector numbers and language codes too!

**Rewrite Error Messages!** "SYNTAX ERROR" can be "TRY AGAIN!" or "NO COMPREHEND"; "DISK FULL" can be "BURP!"... anything you want!

**All of DOS BOSS's change features may be appended to any of your programs, so that anyone using your disks on any Apple (booted or not) will be formatting DOS the way you designed it!**

**Plus the DOS BOSS BOOK!** 36 pages of valuable Apple info! An excellent learning tool covering all DOS BOSS features PLUS a new collection of Beagle Bros. Apple tips & tricks; a great companion to our original Beagle Bros. Apple Tip Book (also included free!).

Inside The DOS BOSS BOOK:

- Discover some strange Apple bugs!
- Put Inverse REM Statements in your listings!
- Two-sided Apple disk tips!
- Make your programs un-listable!
- Custom-format your catalogs!
- Change DOS with creative POKING!

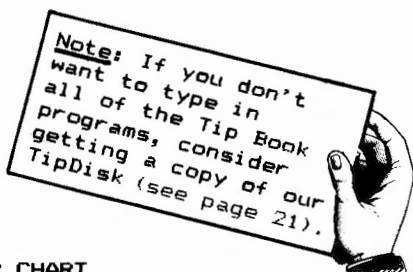


**DOS BOSS ©**  
**and The DOS BOSS BOOK**  
3.2 or 3.3 Applesoft



**Beagle Bros**  
**MICRO SOFTWARE**





#### WEATHER CHART

Everybody's got to have weather, so why not use your Apple to keep a hi-res diary of what's up (and down) in your part of the country. This chart will help you keep track of four months' worth of high and low temperatures as well as daily rainfall. Check the newspaper or collect your data first hand.

Here's what you do; Step 1 is the hardest:

1. Type in both programs and SAVE them.
2. RUN the smaller program. It will ask for which month you want to start the four-month chart.
3. Let your disk drive write a text file for you called WEATHER DATES.
4. LOAD the larger program.
5. EXEC WEATHER DATES. About six pounds of program lines will be automatically added to your program (be glad you didn't have to type them).

LIST 100-, and take a look. Each line of DATA represents a different day. Line 101 would be January 1, Line 1103 would be November 3, etc. Add a Line 229 if this is a leap year. You will update these lines via the main menu each day. The program will halt if it encounters a day with three zeros, so if you want to skip a day, enter three S's as described on the screen. Let us know how everything goes.

=====

#### 4-MONTH WEATHER CHART

=====

```
5 HOME : TEXT : RESTORE : G* = CHR* (7) : Q* = CHR* (
  34) : VTAB 7 : PRINT "WEATHER FOR CITYVILLE" : GOSUB
  90 : PRINT "JANUARY 1, 1985 - APRIL 30, 1985" : GOSUB
  90
7 PRINT "<A> = ADD DATA" : PRINT "<D> = DISPLAY DATA"
  : PRINT "<Q> = QUIT"
10 PRINT : PRINT "SELECT: "; GET A* : PRINT A* : IF A
  * = "D" THEN 60
15 IF A* = "Q" THEN PRINT : PRINT "SAVE WEATHER CH
  ART" : PRINT " IF NECESSARY." : VTAB PEEK (37) -
  2 : END
20 IF A* < > "A" THEN 5
24 REM ===GET DATA===
25 HOME : PRINT "EXAMPLE:" : PRINT "-----
  ----" : PRINT " DATE: NOVEMBER 3" : PRINT " HIGH
  TEMP: 77" : PRINT " LOW TEMP: 58" : PRINT " RAI
  NFALL: .36" : Q* : PRINT
```

```

30 INVERSE : PRINT " ENTER WEATHER DATA AS FOLLOWS:
   ": NORMAL : PRINT : PRINT "1103 DATA 77, 58, 0.3
   6: REM 11/3"
37 PRINT : INVERSE : PRINT " OR, TO SKIP A DAY, ENTE
   R: ": NORMAL : HTAB 9: PRINT "-----": PRINT "1103
   DATA S, S, S: REM 11/3": GOSUB 90
40 FOR I = 1 TO 365: READ H$,L$,R$: IF H$ = "00" AND
   L$ = "00" THEN LINE = PEEK (123) + PEEK (124) *
   256: GOTO 55
45 NEXT
55 PRINT : INVERSE : A$ = "ENTER WEATHER DATA BELOW,
   RUN AND SAVE.": SPEED= 180: PRINT A$;: NORMAL : HTAB
   1: SPEED= 255: PRINT A$
56 M = INT (LINE / 100): IF M > 12 THEN M = M - 12
57 PRINT : PRINT "J ";LINE;" DATA 00, 00, 0.00: REM
   ";M;"/";LINE - INT (LINE / 100) * 100: PRINT : PRINT
   "J SAVE WEATHER CHART": PRINT : PRINT "J RUN": VTAB
   PEEK (37) - 5: PRINT G$;: END
59 REM ===CHART===
60 HGR2 : FOR Y = 0 TO 190 STEP 10: HCOLOR= 2: IF Y -
   20 - INT ((Y - 20) / 50) * 50 = 0 THEN HCOLOR=3
61 HPLLOT 0,Y TO 279,Y: NEXT : HCOLOR= 0: FOR X = 14 TO
   279 STEP 60: HPLLOT X + 1,0 TO X + 1,191: HPLLOT X
   ,0 TO X,191: NEXT
65 X = 15: FOR DA = 1 TO 365: READ H$,L$,R$: IF H$ =
   "00" AND L$ = "00" THEN GET A$: PRINT A$: GOTO 5
70 IF H$ > "9" THEN X = X + 2: GOTO 85
75 H = VAL (H$):L = VAL (L$):R = VAL (R$): IF H =
   0 AND L = 0 AND R = 0 THEN 85
77 IF H > 120 THEN H = 120
78 IF L < - 70 THEN L = - 70
79 REM ===PLOT===
80 H = 120 - H:L = 120 - L:M = L + (H - L) / 2:R = R *
   20: HCOLOR= 5: HPLLOT X,H TO X,M - 1: HCOLOR= 1: HPLLOT
   X,M + 1 TO X,L:X = X + 2
82 IF R THEN HCOLOR= 6: HPLLOT X - 1,190 - R TO X - 1,190
85 NEXT DA
90 FOR I = 1 TO 40: PRINT "-";: NEXT : RETURN
100 REM ===DATA===
101 REM "EXEC WEATHER DATES" TO CREATE DATA LINES.

```

=====

WEATHER DATES WRITER

=====

```

15 D$ = CHR$ (4): PRINT D$;"MONICO"
20 HOME : INPUT "STARTING MONTH (1-12): ";ST: IF ST >
   12 OR ST < 1 THEN 20
30 IF ST > 1 THEN FOR I = 1 TO ST - 1: READ A: NEXT
40 PRINT D$;"OPEN WEATHER DATES": PRINT D$;"DELETE W
   EATHER DATES": PRINT D$;"OPEN WEATHER DATES": PRINT
   D$;"WRITE WEATHER DATES"
50 FOR M = ST TO ST + 3
60 MO = M: IF MO > 12 THEN MO = MO - 12
70 READ L: FOR DAY = 1 TO L
80 PRINT 100 * M + DAY;" DATA00,00,0.00:REM";MO;"/";DAY
90 NEXT : NEXT
100 PRINT D$;"CLOSE"
110 DATA 31,28,31,30,31,30,31,31,30,31,30,31,31,28,31

```

## TWIRLERS

Why bore your audience with an old soggy flashing cursor when you can easily program in something a little bit different!?

```
10 REM
```

### BATON GETTER

```
20 CURS$ = "!/-\" : C = 1 : Q$ = CHR$ (34) : L = LEN (CURS$) :  
    SPEED= 177  
30 HOME : VTAB 10: PRINT "! PRESS ANY KEY WHEN READY."  
40 VTAB 10: PRINT MID$ (CURS$,C,1)  
50 KEY = PEEK ( - 16384) : IF KEY < 128 THEN C = C + 1 -  
    L * (C = L) : GOTO 40  
60 POKE - 16368,0 : VTAB 10: HTAB 3: PRINT "THANK YOU FO  
    R PRESSING THE ";Q$; ( CHR$ (KEY));Q$;" KEY.": VTAB 9  
    : SPEED= 255
```

---

```
10 REM
```

### BATON ROLLER

```
20 CURSR$ = "!/-\" : H = 1 : C = H : M = H : Q$ = CHR$ (34)  
30 HOME : VTAB 11: PRINT " :::::::::: HIT ANY KEY WHEN READ  
    Y. ::::::::::"  
40 VTAB 10: HTAB H: PRINT " "; MID$ (CURSR$,C,1); " ";: FOR  
    I = 1 TO 33: NEXT  
50 KEY = PEEK ( - 16384) : IF KEY > 127 THEN 80  
60 IF M = 1 THEN C = C + 1 - 4 * (C = 4) : H = H + 1 : IF H  
    > 39 THEN M = - 1  
70 IF M = - 1 THEN C = C - 1 + 4 * (C = 1) : H = H - 1 : IF  
    H < 2 THEN M = 1  
75 GOTO 40  
80 POKE - 16368,0 : HOME : VTAB 11: PRINT "THANK YOU FOR  
    PRESSING THE ";Q$; ( CHR$ (KEY));Q$;" KEY."
```

IGNORE THE  
FOLLOWING  
PROGRAM,  
AND TURN  
THE PAGE.



## HEXABET?

I was noticing how some four-letter hex addresses can look like words; like \$BEEF, and \$FACE and \$IDEA. So, I wrote this program to find them all. I took a little artistic license, and let zero be a letter O, and one be an I. You could adapt this program to print every possible four-letter word using the entire alphabet, by changing A\$ in Line 30 to "ABCDEFGHIJKLMNOPQRSTUVWXYZ", and by changing the B's in Line 40 to 26's. It will take quite a while to run however. You COULD speed it up by rejecting all words with three or four vowels or consonants in a row.

```
20 PRINT "HEX DICTIONARY:": PRINT "-----"  
30 A$ = "ABCDEF10"  
40 FOR A = 1 TO 8: FOR B = 1 TO 8: FOR C = 1 TO 8: FOR D  
    = 1 TO 8: PRINT MID$ (A$,A,1); MID$ (A$,B,1); MID$  
    (A$,C,1); MID$ (A$,D,1); " ";  
50 NEXT : NEXT : PRINT : NEXT : PRINT : NEXT
```



### PLEASE PARS THE VARIABLES

Applesoft can do some weird things with variables. For instance, the statement,

IF X = T AND Y = 0 THEN PRINT  
will be respaced or "parsed" to read,

IF X = TAN DY = 0 THEN PRINT  
and cause a ?SYNTAX ERROR, because TAN is a reserved word meaning tangent. To prevent this, rename the variable or put it in parentheses:

IF X = (T) AND Y = 0 THEN PRINT  
More details on parsing are somewhere in your Applesoft Manual. I don't have time right now to check the page number.

### INSIGNIFICA

- > You can't boot from Drive #2.
- > Escape-@ clears the text screen and leaves you with a cursor, but NO PROMPT.
- > You can use a GOSUB as a direct keyboard command; good for testing a subroutine.
- > "PR36" won't boot a disk.
- > "NEXT:NEXT" uses less memory than "NEXT X,Y" and might even be faster (check it and let us know).
- > Everything in a program line after an ONERR statement is ignored by Applesoft.
- > Everything in a program line after a REM statement is ignored by Applesoft.
- > Everything in a program line after a GOTO statement is ignored by Applesoft.
- > A language card has to go in Slot #0, but other cards may go anywhere. Traditionally, printers are connected to Slot #1 and disk drives are connected to Slot #6 and Slot #5.



### PRE-BOOT NEWS

You can turn your Apple on or do a PR#6 BEFORE inserting a disk. It's reasonably safe to insert a disk into a drive that is running...

...I think.

### TO JUSTIFY OR NOT TO JUSTIFY

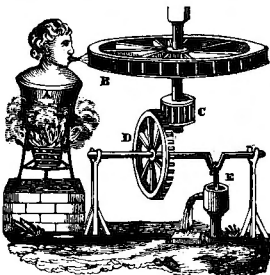
Computerers with printers get all excited about justified type. I don't, as you can see by the text format in this Tip Book (printed on my Epson MX-80, by the way). Justified type is attractive if the extra spacing necessary is done evenly and in very small increments, but lower-cost printers add space only between words, usually all on the left side of the paragraph, and don't really produce super attractive results. These two paragraphs make a fair comparison. What do you think?

### TO JUSTIFY OR NOT TO JUSTIFY

Computerers with printers get all excited about justified type. I don't, as you can see by the text format in this Tip Book (printed on my Epson MX-80, by the way). Justified type is attractive if the extra spacing necessary is done evenly and in very small increments, but lower-cost printers add space only between words, usually all on the left side of the paragraph, and don't really produce super attractive results. These two paragraphs make a fair comparison. What do you think?

### INVERSE INIT

You can INIT a disk with an inverse greeting program name, but you can't Master Create it without changing the name to normal text.



### WORKING(?)

Many programs use a flashing "WORKING" note to tell you to stand by while the computer does some task of indeterminate length (sorting, for example). The problem is that for all you know, the program is frozen and will NEVER finish. A better "working" symbol would be one that slows down or stops when the program slows down or stops; something like:

```
999 HOME
1000 W = W + 1: VTAB 10: HTAB 1 + (W - INT (W / 2) * 2 =
    0): PRINT SPC( 1);"WORKING"; SPC( 1)
1010 REM (working subroutine here)
1020 FOR I = 1 TO 250: NEXT
1030 IF W < 15 THEN 1000
1040 VTAB 10: HTAB 1: PRINT "DONE"; CALL - 958: VTAB 20
```

The "working" routine of your program should encounter line 1000 repeatedly. Line 1020 is for demo purposes only.

### THE MYSTERIOUS COLOR BIT

You may have wondered if there is a difference between the colors BLACK #0 & BLACK #4 and WHITE #3 & WHITE #7. Well, here's an experiment that proves these different colors really do exist:

First, get Alpha Plot running and CLEAR THE SCREEN in a color OTHER THAN WHITE OR BLACK. Now, switch pages with Option 6-S and CLEAR THE SCREEN IN COLOR #0, BLACK. You now have a solid color on Page 2 and a black screen on Page 1.

Now, SET THE COLOR TO 4 (the other black). At various places on the screen, draw some filled boxes and ellipses. Since you are drawing black on black, they won't look like much, but trust me...

Here's the fun part-- Select Option 6-M for Merge. Select Merge Option #2, Xdraw Page 1 onto 2. Now look at Page 2. Your invisible figures have APPEARED! If you pick the same merge option again, they will disappear before your very eyes!

Now try this-- Go back to Drawing Mode on Page 1. Use the Rubber Band Cursor, and line it up vertically over some of your invisible figures. Wherever the Rubber Band intersects your shapes, it will take a little sideways step and change colors. Once again, you can see the reality of the invisible color bit!



### **Beagle Bros TipDisk**

Hey out there! Don't type in all of those programs from the Beagle Bros Tip Books! Here they are--typed, tested and ready to run--ALL OF THE PROGRAM LISTINGS FROM TIP BOOKS 1, 2, 3 AND 4! Many are useful; some are useless; ALL are interesting, listable and copyable. And each program teaches another elusive fact about making your Apple do its thing. No one around here has had time to count them all, but you can figure around four times as many programs as you have here in this Tip Book. The Two-Liners too, from all over the world (and elsewhere). Order now, and you'll have your disk in a few days!

**BEAGLE BROS TIPDISK #1: \$20.00**



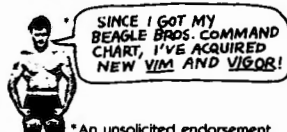
# ONERR TELL ME

ONERR GOSUB and ONERR GOTO are great for keeping a program on it's feet and running when an error is encountered. But you want to CATCH those errors if you are programming. You should either take the ONERR statement temporarily out of your programs or put Line 1000 (below) in. Actually, this line could be put into ALL of your programs.

```

10 G*=CHR$(7): ONERR GOTO 1000
20 X=X+1: NORMAL: PRINT X,
30 IF X = 19 THEN PRITN "NINETEEN": REM
   INTENTIONAL ERROR
40 IF X = 30 THEN GOTO 99: REM
   INTENTIONAL ERROR
50 IF X<35 THEN 20
60 END
1000 INVERSE: PRINT G*;" ERROR #";PEEK(222);" ON
      LINE ";PEEK(218)+PEEK(219)*256;".": GOTO 20
  
```

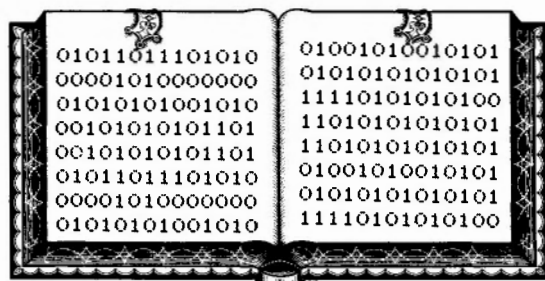
Run the program and compare the error numbers with the ONERR codes in your Applesoft Manual.



\*An unsolicited endorsement

# A \$2.50 FREEBEE

We've retired the old Apple Command Chart as the head Beagle Bros freebee, and replaced it with our Peeks & Pokes Chart. We've still got plenty of Command Charts in the attic though, really nice too-- a handy alphabetical list of 190 Applesoft, Integer and DOS commands. If you want one, send us \$2.50 + 75 cents postage & handling (don't you hate it when mail order places do that?), and we'll send you a Command Chart right away.



### HI-RES PAGE THREE!

I don't know if Apple calls it Page Three, but I do. I discovered by accident that you can store a full-screen Hi-Res image above Page Two, and move it to one of the viewing pages (Page One or Page Two) anytime you want. There's a pointer at location 230 (\$E6) that tells you what page you are DRAWING on. Use the following pokes to change the drawing page:

POKE 230,32 (Draw on Page One)

POKE 230,64 (Draw on Page Two)

POKE 230,96 (Draw on Page Three)

The page you are LOOKING at, One or Two, is determined by HGR or HGR2, whichever is more recent, or by the display switches at -16299 and -16300:

POKE -16300,0 (Look at Page One)

POKE -16299,0 (Look at Page Two)

Now, let's try a test. Type in the following program:

10 HGR2: HGR: HCOLOR=3: POKE 230,96:

REM SET-UP DRAW ON PAGE 3

20 H\$PLOT 0,0 TO 279,191: H\$PLOT 0,191 TO 279,0:

REM DRAW AN X (WON'T SHOW)

30 VTAB 21:END

You just drew a big X on Page Three, and you're looking at a blank Page One. Page Two is blank as well. To move the X to Page One so we can see it, enter the monitor and move the picture in memory, like so:

CALL-151 (return)

2000<6000.7FFFF (return)

Page One resides in memory locations \$2000 to \$3FFF (8192 to 16383). Page Two is \$4000 to \$5FFF (16384 to 24575). Page Three is \$6000 to \$7FFF (24576 to 32766). Knowing this, you can move an image from one page to another by altering the numbers in the above move command. Remember, however, that to LOOK at a certain page, you have to throw the -16300 or -16299 switch, and you can't look at Page Three.

I guess I should mention... "Page Zero" in Hi-Res terminology would encompass memory locations \$0 through \$1FFF, which includes the real Page Zero (\$00-\$FF). "Drawing" on this page can be disastrous. This is exactly what happens if you turn on your Apple and start hplotting without doing an HGR or POKE 230,32. Because the value at location 230 is zero, you overwrite Page Zero and really make a mess. Likewise, "Page Four" is unusable, because that's where DOS is. Don't draw on DOS!





### TELETYPER

Instead of having type just appear on the screen, why not give it some importance with a fancy printing cursor and a little noise. It seems like computers are SUPPOSED to do this! Insert your own copy in Line 50.

```

20 CURSR$ = ">"; SPEED= 210
30 FOR I = 1 TO LEN (CURSR$):CURSR$ = CURSR$ + CHR$ (8
   ): NEXT
40 HOME : VTAB 7: GOSUB 100
50 A$ = "ONE DAY, IN THE MIDDLE OF THE NIGHT, TWO DEAD BOY
   S GOT UP TO FIGHT. BACK TO BACK, THEY FACED EACH OTHE
   R; DREW THEIR SWORDS AND SHOT EACH OTHER. A DEAF POL
   ICEMAN HEARD THE NOISE, AND CAME AND KILLED THOS
   E TWO DEAD BOYS.": GOSUB 70
60 PRINT : GOSUB 100: SPEED= 255: VTAB 22: END
70 FOR I = 1 TO LEN (A$): INVERSE : PRINT CURSR$: NORMAL
   M$ = MID$ (A$,I,1): PRINT M$: IF M$ = " " OR M$ =
   "." THEN FOR J = 1 TO 25 + 150 * (M$ = "."): NEXT
80 IF M$ < > " " THEN FOR J = 1 TO 2: S = PEEK ( - 163
   36): NEXT
90 NEXT : CALL - 868: PRINT : RETURN
100 A$ = "": FOR I = 1 TO 40: A$ = A$ + "-": NEXT : GOSUB
   70: RETURN
  
```

### GET PROMPT

PRINT PEEK(51). Better yet, PRINT CHR\$(PEEK(51)) if you're using Applesoft. This little exercise will show you what language you are using (in case you didn't know!) by printing its prompt character. Integer's ">" is represented at location 51 in memory by a value of 190. Applesoft's "I" is a 221. Try to POKE in a different number... Forget it; you can't. Another location that tells your language is 43702 in DOS (27318 if you've got 32K). PRINT PEEK(43702). If you're in Integer, you'll get a zero. Applesoft ROM gives you a 64; Applesoft RAM a 128.



### PLAY IT SAFE

SAVE each program you type before you RUN it. Sometimes a bug or typo will cause a program to self-destruct when it's run.

# utility city

21 of our most-asked-for Apple Utilities  
by Bert Kersey

48K Applesoft ROM

For YOUR Big Apple—Our 21 most-asked-for Apple Utilities on one big disk—List Formatter makes properly spaced & indented listings with printer page breaks. Each program statement is on a new line with If-Then's & Loops called out; a great de-bugger! Catalog in any number of columns & any page-width to CRT or printer. Automatically post the Run-Number & last-used Date in your programs. Make any command invisible in your listings; Access program lines in memory for garbage repair & "illegal" alteration; Quickly sort & store info on disk; Run any Applesoft file while another stays intact; Renumber to 65535; Save inverse, INVISIBLE & trick file names; Convert dec to hex & binary, or INT to IP; Append programs; dump the text screen to ANY printer. . . More too: 21 Programs Total!

## PLUS . . . APPLE TIP BOOK NUMBER THREE!

Peek's, Pokes, & Pointers Chart, Apple's new keyboard shortcuts, & more. . . All the tips you need to know to get the most out of your Apple. . . All the tips you need to know to get the most out of your Apple. . . All the tips you need to know to get the most out of your Apple. . .

- ✓ **UTILITY CITY** on Applesoft Disk.
- ✓ Beagle Bros Apple Tip Book #3
- ✓ Apple PEEKS, POKES & POINTERS Chart

## NORMAL LISTING:

```
530 SCALE= J: ROT= 32 & (X2 > XN
  ): XDRAW 4 AT XN,YN: XDRAW 4
  AT XN,Y2: SCALE= ABS (YN -
  Y2) - 1 + 2 & (YN = Y2): ROT=
  16 + 32 & (Y2 > YN): XDRAW 4
  AT XN,YN + 1 - 2 & (YN > Y2
  ): XDRAW 4 AT X2,YN + 1 - 2 &
  (YN > Y2): ROT= R: SCALE= 1
535 IF ABS (XN - X2) > 255 THEN
  SCALE= ABS (XN - X2) - 254
  : ROT= 0 + 32 & (XN > X2): XDRAW
  4 AT X2,YN: XDRAW 4 AT X2,Y2
560 HPLLOT X2,YN TO XN,YN TO XN,Y
  2 TO X2,Y2 TO X2,YN: IF C =
  4 OR C = 7 THEN J = X2 + 1 -
  (X2 = 279): HPLLOT J,YN TO J,
  Y2: J = XN + 1 - (XN = 279): HPLLOT
  J,YN TO J,Y2
```

## RE-FORMATTED LISTING:

```
530 SCALE= J
  : ROT= 32 & (X2 > XN)
  : XDRAW 4 AT XN,YN
  : XDRAW 4 AT XN,Y2
  : SCALE= ABS (YN - Y2) - 1 + 2 &
    (YN = Y2)
  : ROT= 16 + 32 & (Y2 > YN)
  : XDRAW 4 AT XN,YN + 1 - 2 & (YN > Y2)
  : XDRAW 4 AT X2,YN + 1 - 2 & (YN > Y2)
  : ROT= R
  : SCALE= 1
535 IF ABS (XN - X2) > 255 THEN
  SCALE= ABS (XN - X2) - 254
  : ROT= 0 + 32 & (XN > X2)
  : XDRAW 4 AT X2,YN
  : XDRAW 4 AT X2,Y2
560 HPLLOT X2,YN TO XN,YN TO XN,Y2 TO
  X2,Y2 TO X2,YN
  : IF C = 4 OR C = 7 THEN J = X2 +
    1 - (X2 = 279)
  : HPLLOT J,YN TO J,Y2
  : J = XN + 1 - (XN = 279)
  : HPLLOT J,YN TO J,Y2
```

UTILITY CITY is a huge collection of Applesoft utilities that will keep you off of the streets for a LONG time. My favorite is XLISTER, a listing printout routine, demonstrated at the left. Everything is properly spaced; Each program statement appears on a new line; If-thens are called out, and. Loops are indented. Here's the entire list of U-City programs, in no particular order:

## FILENAME ZAP:

Invisible, inverse & trick file names

## MULTI-CAT:

Multi-column catalogs to any-width printer

## SCREENWRITER:

Layout & format text directly on the screen.

## XLISTER:

A properly-spaced lister. See above.

## COMMAND ZAP:

Puts invisible commands in your programs.

## DOUBLE LOADER:

Run a file while another stays intact.

## HEX, DEC & DEX:

Hex conversions while your program stays intact.

## INT CONVERTER:

Convert Integer BASIC to Applesoft.

## KEY-CAT:

1-key program selection (also on Dos Boss).

## LINE SEARCH:

Locates program lines for "illegal" alteration.

## BIGLINER:

Renumber to 65535 for program protection.

## CONNECT:

Append two Applesoft programs to make one.

## KILL-CAT:

Allows a ctrl-C clean escape from long catalogs.

## RUN/DATE COUNTER:

Posts an updated run number or date when run.

## TEXT DUMP:

Send your text screen to any printer.

## ADDRESS CHECKER:

Learn while you snoop through memory.

## BFIND:

Finds binary program start address and length.

## SORTFILE:

Sort, store and manipulate data on disk.

## CHR\$ POKER:

Finds text screen locations.

## REM ZAP:

Puts invisible rem statements in your listings

## REM FIND:

Converts zapped rems to temporary INVERSE

# **Alpha Plot Instructions**

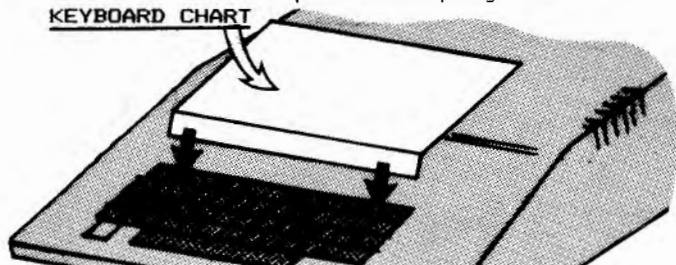
## **General Information**

Alpha Plot is a sophisticated Hi-Res graphics program that will let you flex your Apple's Hi-Res capabilities to create colorful drawings and detailed charts and graphs. These instructions will do their best job if you run Alpha Plot and experiment while you read. Most of the details not in the instructions are printed on the screen.

### **ALPHA INFO**

Any changes or suggestions that we have come up with since this book was printed appear on the Alpha Plot disk. Run the "Alpha Info" program.

### **KEYBOARD CHART**



Insert the keyboard chart that came with the disk behind the top row of keys on your Apple. The chart keeps all Alpha Plot commands in front of you and prevents the need for remembering the many key codes. All but a few Alpha Plot functions are initiated by pressing a top-row key.

### **ALPHA PLOT'S TWO MODES**

Alpha Plot features a Drawing Mode and a Typing Mode. To enter either mode, select Key Chart Option 4, then D or T, OR type ctrl-D or ctrl-T while typing or drawing. In the Drawing Mode, you can do everything EXCEPT type. Typing Mode lets you type and access Options 1-6.

### **TWO IMAGE PAGES**

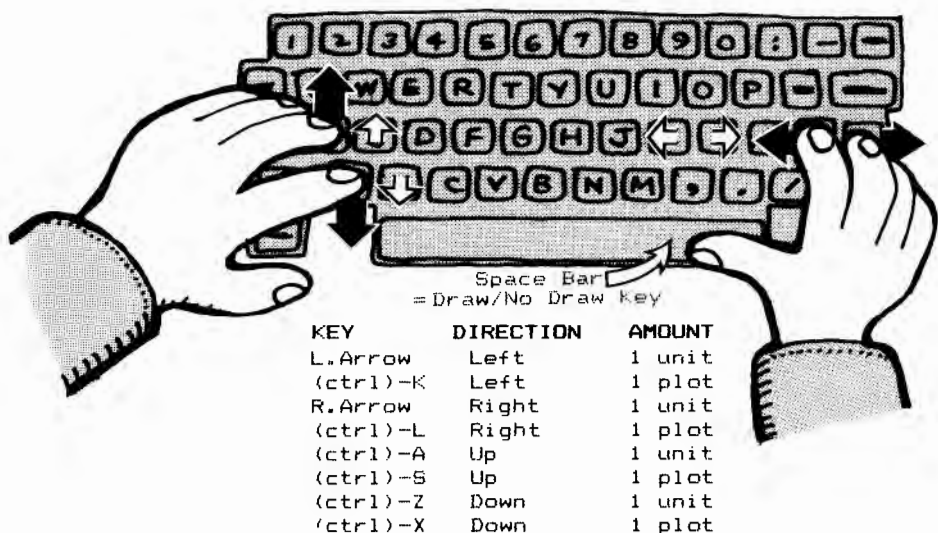
The Apple has two Hi-Resolution screens or "Pages", called Page 1 and Page 2. With Alpha Plot, you can store two images in memory at the same time and view either one. Alpha Plot lets you draw or type only on Page 1, but you may easily move images from Page 1 to Page 2 and back, and from memory to disk and back from or to either page. You may, of course, store as many images as you want on disk.

### **APPLE'S SCREEN LAYOUT**

Apple's Hi-Res screens measure 280 plots wide (0-279) by 192 plots high (0-191). Four colors (1,2,5,6), two blacks (0,4) and two whites (3,7) are available for Hi-Res plotting. See "<1> Color Selection" and your Applesoft Manual and for more information.

## CURSOR MOVES

Cursor moves from the Drawing and Typing Modes are similar. From the Typing Mode, you must press the CTRL key WHILE you press the appropriate cursor-move key. Otherwise, you would type the letter on the screen.



See more about units, plots, and other cursor-move features in the Typing Mode and Drawing Mode sections. The Alpha Plot key layout works well for comfortable two-hand operation. Notice that the plot-move keys are adjacent to the unit-move keys.

## <RETURN>

If you find yourself in a part of Alpha Plot you don't want to be in (for example, you might accidentally press a 5 and be in the "Erase the Screen" mode), pressing the <RETURN> key as an answer to a question will usually get you back to the Drawing or Typing mode.

## COMPATIBILITY WITH NORMAL DISKS

Images created with Alpha Plot are completely compatible with your normal programs and disks. Therefore, you may use pictures made with Alpha Plot in your own programs. Run and list the program on the Alpha Plot disk called "PICTURE BLOOD DEMO" if you are unfamiliar with loading and viewing hi-res pictures under program control.

## SAVE TO BE SAFE

Making a drawing on your Apple is quite a bit like programming. The more you plan, the better things turn out. The more you practice, the more efficient you will be. As with programming, SAVE YOUR WORK from time to time in case you want to continue drawing from where you were ten minutes ago (or in case the dog pulls the plug).

## Starting Out

Boot the Alpha Plot disk, and choose the "Run Alpha Plot" option. Answer Y to "Clear the Hi-Res screen?" (or answer N if you have pictures in memory that you want to keep). After the usual amount of disk-whirring, you will be presented with Hi-Res Page 1, with text at the bottom of the screen offering you the choices D, T, L and S for Draw, Type, Load and Save. You will re-encounter this portion of the program each time you press the 4 key. More later about that.

## Drawing Mode

From this mode you can do anything except type-- plot lines, dots, ellipses and boxes, and perform page and image manipulations.

### MOVING THE CURSOR

When you first run Alpha Plot, you will see a flickering squarish dot on the screen. This is the XO CURSOR, actually four cursors in one. Press the A, Z and Arrow keys a few times to separate the four cursors and to get a feel for cursor movement. Try the K, L, S and X keys too for one-plot movement. Notice how the O part of the cursor is controlled by you while The X stays in one position, and the two dots go along for the ride.

### CURSOR JUMPS

How far your cursor goes with each keypress is determined by the SPEED (9 key). However, you may JUMP the cursor 50 PLOTS without changing the speed. Simply press <ESC>, then A, Z or an Arrow. To jump 150 plots down from the top of the screen, for example, type <ESC>-Z-<ESC>-Z-<ESC>-Z.

### CURSOR COORDINATES and OFFSET

At the bottom left of the screen is information regarding the current position of the cursor. The H and V values after the word "CURSOR" represent the horizontal and vertical coordinates of the O part of the cursor. The "OFFSET" values represent the horizontal and vertical distances BETWEEN the X and the O. Notice that coordinates H=0 and V=0 place the cursor at the upper left of the screen and coordinates H=279 and V=191 place it at the lower right.

**NOTE:** To exit the Drawing Mode, type "4" or ctrl-T.

### THE KEYS

Except for cursor moves, most Alpha Plot features are controlled by the top row of keys as indicated on your Keyboard Chart. Here is a discussion of each key and what it does.

## (Space Bar) Draw/Move Selection

Press the space bar and you will see the flashing word, "DRAW", on the top left text line. Now move your cursor and it will indeed draw. Press the space bar again and "DRAW" will change to "MOVE", meaning you can move the cursor without drawing. The space bar is used because it

is easily reachable with your thumb while you are manipulating the cursor. To plot a single dot, press the space bar twice and move the cursor away.

## **(Return) Zero Offset**

Pressing the <Return> key while you are in the Drawing Mode will set the offset to zero and place all four corners of the XO Cursor on top of the O. This same feature is accomplished by pressing 8 and Z. See "<8> Modify Cursor".

## **(1) Color**

Press the 1 key and you can select your Drawing or Typing (size>1) color. If Alpha Plot isn't drawing or typing when you think it should be, maybe you have the drawing color set the same as your background. To select a new color quickly, simply type 1-3 or 1-6 or 1-whatever, depending on the color you want.

### **COLOR BUGS**

1. The Apple won't let you draw certain colors in certain vertical columns. You can only draw in Colors 1 and 5 in ODD-NUMBERED columns. Colors 2 and 6 only appear in EVEN-NUMBERED columns. Alpha Plot corrects this by moving the cursor one plot over when necessary.

2. A WHITE VERTICAL line will appear in color unless you plot another line directly next to it. Alpha Plot does this for you for Colors 4 (Black) and 7 (White), but NOT for Colors 0 (Black) and 3 (White). That way you have a choice when drawing vertical lines.

### **SPECIAL COLORS: 8 & 9**

COLOR #8 (REVERSE) lets you draw lines or boxes (not ellipses) in the OPPOSITE of the background color. These lines and boxes may be erased by re-drawing over them in Reverse. Experiment and see.

COLOR #9 (MIX) lets you select TWO COLORS (not Reverse) for drawing filled boxes and ellipses. Every other horizontal line will appear in each color. You can create many different shades and hues by mixing colors (and by adjusting a few of the knobs on your tv set!). Load the "Color Chart" picture from the Alpha Plot disk to see the range of color possibilities. If your color is #9 (mix), your drawing color for lines will be the FIRST COLOR of the mix.

## **(2) Notes**

The notes at the bottom of the screen may be turned off to reveal whatever image may be behind the notes. Press 2 for full-screen graphics or to turn the notes back on again.

## **(3) Grid**

Press 3 to draw a Hi-Res grid over your picture. Press 3 again and the grid will be erased (that's Color #8 again!). There are dots at every ten plots in both directions. The grid lines are 100 plots apart. It is best NOT to draw or type with the grid on the screen

unless you want the grid as a permanent part of your picture. When the grid erases, it will draw little dots and stripes on any NEW parts of your drawing. Another way to access the grid is to draw it on Page 1 and store it on Page 2 (see Option 6).

## **(4) Draw, Type, Load & Save**

Running Alpha Plot or selecting Key Chart Option 4 will get you into the "Draw, Type, Load, Save" mode with its animated text arrow. The Drawing Mode and Typing Modes (described earlier) may be entered from this option by selecting "D" or "T".

"Load" and "Save", selectable by pressing the L or S keys, let you load and save (actually BLOAD and BSAVE) hi-res pictures to and from disk. Hi-res pictures will usually appear as 34 sector binary (B) files in your catalogs. Scrunched picture files (see page 38) will be smaller.

You may catalog whatever disk is in your drive by first selecting L or S and then typing "CAT" (return) when a file name is requested. You cannot perform other DOS functions such as Lock and Unlock while using Alpha Plot. To resave a picture that is locked, save it with a different name OR save it onto another disk. If you want, you can even quit Alpha Plot and load and save hi-res images "by hand" by typing:

```
BLOAD PICTURE,A*2000      (page 1)
BLOAD PICTURE,A*4000      (page 2)
BSAVE PICTURE,A*2000,L*2000 (page 1)
BSAVE PICTURE,A*4000,L*2000 (page 2)
```

The "PICTURE BLOAD DEMO" program demonstrates more regarding handling of hi-res pictures.

### **ERRORS**

If the disk is full or some other error occurs during a Load or Save, you will see an error message number on the screen and the program will continue. The error number represents the TYPE of error that has occurred. Here are the most likely numbers: 4=WRITE PROTECTED, 6=FILE NOT FOUND, 8=I/O ERROR, 9=DISK FULL, 10=FILE LOCKED, 13=FILE TYPE MISMATCH. See your Applesoft Manual for the complete list. If you are attempting to save an image and get a DOS error, such as "File Type Mismatch" or "Disk Full", your picture will NOT BE SAVED completely. You will need to exit Alpha Plot, correct the problem, Run Alpha Plot, and save your image again.

## **(5) Clear the Screen**

Press 5 and select the color you want to clear the screen. Your picture will be permanently erased unless you pick "Color" #8 which will give you a NEGATIVE IMAGE of the screen. "Erase" again with Color #8 and you'll get your positive image back!

## (6) Image Options

Pressing 6 gives you four powerful options:

### (6-P) SEE PAGE 2:

Press P, and you will see your Page 2 image (if there is one). Now press any key, and the screen will flicker between Pages 1 and 2 so you may compare them. Press any key again, and you are back to Page 1.

### (6-S) SWITCH IMAGES:

Press S and your two pages will be switched. Now you can draw or type on your (former) Page 2 image.

### (6-R) RELOCATE:

Any rectangular section of Page 1 may be duplicated on either page within the limits of the screen. BEFORE YOU SELECT OPTION 6-R, define the area to be moved with the four points of the XO cursor. Now press 6 and R, select 1 or 2 for the page you want the move made to and move the flashing rectangle to the desired new location. The 2 key will function here if you want to temporarily remove the screen notes. PRESS M TO MAKE THE TRANSFER or <Return> to escape. After the transfer has been made, you may delete the original section image by drawing a solid box over it.

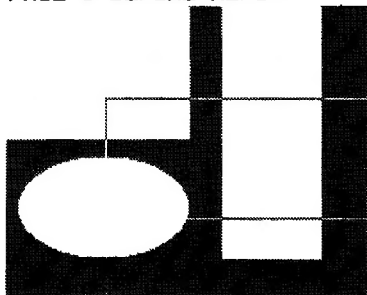
### (6-M) MERGE PAGES:

Alpha Plot lets you merge pictures four different ways. If you're into the terminology ("A OR B", etc.), here it is. If you're not, ignore this:

1. A OR B (Opaque Non-Black Page 1 onto Page 2)
2. A EOR B (Xdraw Page 1 onto Page 2)
3. A AND B (Opaque Black Page 1 onto Page 2)
4. A=B (Compare/Combine)

You will want to experiment with each option. If your Page 2 picture is valuable, SAVE IT on disk BEFORE MERGING. Here are two sample images. The four possible merges appear on the next page.

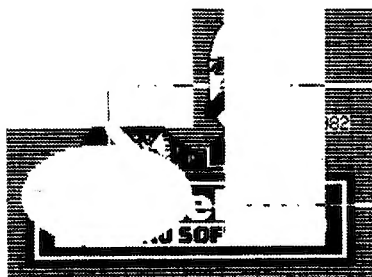
PAGE 1 BEFORE MERGE



PAGE 2 BEFORE MERGE







PG.2 AFTER MERGE OPTION 1



PG.2 AFTER MERGE OPTION 2



PG.2 AFTER MERGE OPTION 3



PG.2 AFTER MERGE OPTION 4

## (7) Keyboard/Paddle Switch

The 7 key will switch cursor control from paddles to keyboard and back with a "KEY" or "PDL" appearing on the top line of the screen notes. If you don't have paddles, you don't need paddles; keyboard control is far more accurate. Paddle-0 will move the drawing cursor horizontally and Paddle-1 will move it vertically. Paddles have no effect in the Typing Mode. All keys except the cursor-move keys have their normal effect with paddles in control.

## (8) Modify Cursor

The 8 key lets you select or re-arrange the cursor. The current cursor's symbol; "XO", "RB" or "---" will appear in the screen notes.

### (8-X) XO CURSOR

This is the Drawing cursor you will probably use most, since it is the most versatile of the three available. The XO cursor consists of four points. The O is the drawing point, the X is a stationary reference point, and the other two points assist in the drawing of ellipses, boxes and lines. Pressing <Return> (or Option 8-Z) or drawing a line (Option "-") will put all four points on top of the O.

### (8-B) RUBBER BAND CURSOR

The Rubber Band cursor works similarly to the XO cursor, but shows you a stretchable projected line between a stationary point and any other point you select. Using this cursor, you can actually see a line plot before you draw it. Pressing <Return> (or Option 8-Z) or drawing a line (Option "-") will give the

Rubber Band a length of zero.

#### **(8-N) NO CURSOR**

The no-cursor mode is used when you want to draw with no distractions on the screen. With no cursor you can do everything you can with either of the two visible cursors.

#### **(8-Z) ZERO OFFSET**

Z will put all four XO Cursor points on top of the current O position, or make the length of the Rubber Band Cursor zero. Select Z when you want to move the usually stationary X position. Pressing <Return> from the Drawing Mode is the same as selecting 8-Z.

#### **(8-R) ROTATE LEFT**

R rotates the XO Cursor points or the Rubber Band 90 degrees counter-clockwise (unpredictable in PDL mode).

### **(9) Speed**

Speed determines the NUMBER OF PLOTS the cursor will plot or move for each A, Z or Arrow keypress. You may select speeds 1-9 by pressing the appropriate number. Pressing a zero will select a speed of 10. "\*" is 20 and "-" is 40. S, X, K and L will move the cursor ONE PLOT regardless of the speed. <ESC> plus A, Z, or an Arrow key will move the cursor 50 PLOTS regardless of the speed. To quickly select a new speed, simply type 9-5 or 9-\* or 9-whatever.

### **(0) Ellipse**

Pressing the Zero key will draw an ellipse that would touch each of the four sides of an imaginary box connecting the four XO cursor points. Another Zero will fill the ellipse. You do not have to wait for the circle to be drawn to fill it (simply type two Zeros). You may halt an ellipse while it is being drawn by pressing <Return>. An ellipse may be any color or color mix, but not Reverse. Theoretically, to draw a CIRCLE, the H and V Offset values at the bottom of your screen should be about EQUAL. This depends partly on the amount of distortion on your monitor. Try a test.

### **(:) Box**

Pressing a Colon will draw a linear box connecting the four points of the XO cursor. Colors 4 and 7 black or white boxes have fatter vertical sides. Colors 0 and 3 do not (see "Color Bugs"). A second Colon will fill the box in the selected color or Reverse. You may use mixed colors for drawing boxes (see more under Color Selection).

### **(-) Line**

Pressing a Minus-sign key will connect the X and O ends of your XO cursor with a line or make an imprint of the Rubber Band Cursor on the screen in the appropriate color. The offset will be set to zero at the O end of the cursor. To erase a Reverse line, simply draw over it in Color #8.

## **(Reset) Quit**

Hitting Reset OR CTRL-@ will give you the option of exiting Alpha Plot. Note: With a reset exit, you risk imprinting the cursor on your picture; no problem if you have already saved your picture. Alpha Plot erases itself when you quit. To list it, LOAD it, then LIST.

-----

## **Typing Mode**

Alpha Plot lets you type directly onto Page 1 using the normal set of ASCII characters. There are a few characteristics you will notice that make Alpha Plot's type different from the type you are used to seeing on your Apple screen--

### **LOWER CASE:**

Whether or not you have lower-case hardware installed in your Apple, you can type in upper and lower case with Alpha Plot. The ESC key controls upper and lower case.

### **SAME-LINE WRAPAROUND:**

When you reach the edge of the screen while typing in any direction, the type will jump to the opposite margin, but stay on the SAME LINE. Use the carriage return when you reach the edge of the page (as you would on a typewriter) if you don't want this to happen.

### **SCREEN POSITION:**

Any character may appear at any vertical or horizontal screen position. There is no connection between Alpha Plot's type and Apple's HTABS and VTABS. Therefore, you may adjust the distances between lines and characters, and make vertical adjustments for subscripts, superscripts and so on.

### **PROPORTIONAL SPACING:**

With Alpha Plot text, each letter only takes up as much horizontal space as is necessary. Most characters are the normal five plots wide. Some characters, like I's l's and f's, are narrower. Some, like W's and M's, are wider. Proportional type is attractive and easy to read, and you can fit more characters per line on the screen than with normal type. The only disadvantage comes in backspacing, and vertical character alignment; a couple of extra keystrokes are sometimes necessary to align cursors and characters. Which brings us to--

### **CURSOR MOVEMENT**

To move the cursor in the typing mode, you must HOLD DOWN THE CTRL KEY while you press the cursor-move keys. CTRL-A and CTRL-Z move the cursor up and down one the height of one line of type (plus leading). The left and right arrows (ctrl-optional) move the cursor left and right the width of a normal character (7 plots at normal type size and kerning). CTRL-S, CTRL-X, CTRL-K and CTRL-L move ONE PLOT up, down, left and right, allowing you to move to any precise spot on the screen.

**NOTE:** To EXIT the Typing Mode, hit ctrl-D to draw, or hit ctrl-O and "4" to draw, load or save.

#### **CTRL-J: ERASE A SEGMENT**

Ctrl-J will erase a letter segment one-plot wide and one-character high. You must follow each ctrl-J with a ctrl-K or ctrl-L to advance the cursor left or right.

#### **<ESC> UPPER/LOWER CASE**

The ESC key will set the upper & lower case switch in three different positions; 1:Lower Case Only, 2:First Character Upper Case and 3:Upper Case Only. Position 2 will produce a capital letter on the next keystroke and then switch to lower case operation. The height of the flashing cursor and the notes at the bottom of the screen will tell you whether the next character typed will be upper or lower case.

#### **<CTRL-O> OPTIONS**

Ctrl-O (letter O) will let you select typing options or exit the Typing Mode.

##### **(Ctrl-O,1-6)**

Drawing Options 1-6 (on your Keyboard Chart) are available from the Typing Mode AFTER TYPING CTRL-O.

##### **(Ctrl-O,B) BIG COLOR (same as Option 1)**

You may select the color for type larger than size 1. Due to the color dot layout of Apple's graphics system, normal-sized type in color is not readable, so it's not available on Alpha Plot.

##### **(Ctrl-O,M) MODE**

M changes the type output mode from Inverse to Normal to Xtype and back. Each Inverse and Normal character clears a path for itself, so you will always have white or black type on the opposite color. XTYPE will let you type over your drawings with the type changing color according to the background. Color type of size #1 is illegible, so don't use Xtype over a color background in size 1. Xtype is also sometimes hard to read when used over complicated backgrounds.

##### **(Ctrl-O,R) ROTATE TYPING CURSOR**

R allows you to type sideways (good for labeling graphs, etc.) and upside down (good for people who are upside down). The cursor points the direction of the type. Carriage returns and cursor move characteristics are rotated with the cursor.

##### **(Ctrl-O,S) TYPE SIZE**

You may type in four type sizes. Every time you press S, the type size will increase up to type-size #4. Then it will go back to #1 or normal-size type. You will need to type a bit slower with type larger than size 1.

#### **(Ctrl-O,L) LEAD**

L controls the vertical distance the cursor is moved for each carriage return, and thus determines the leading or space between type lines. Normal leading on the Apple is 1 (and somewhat hard to read). Because of Alpha Plot's lower case descenders (parts of letters like g's and y's that go beneath the base line), an extra space of leading is preferable, so 2 is standard.

#### **(Ctrl-O,K) KERN**

K lets you adjust the kerning or space between characters on the screen. Normal is 2, but adjust it to suit your preference, to squeeze more type into a small space or to stretch a line out to make a title or whatever. On a color monitor, type kerned one unit could be difficult to read.

---

## **Possible Problems**

### **NO CURSOR**

The Drawing or Typing cursor can get hidden behind the screen notes. Try moving the cursor up, or select Option 2 (or ctrl-O,2) to remove the notes. You will also have no cursor if you are in one of the Option modes (top row of keys). Try hitting <Return>.

### **THICK VERTICAL LINES**

Use black or white colors 0 or 4 instead of #3 or #7.

### **NO PLOT OR TYPE**

Check your plotting color. It could be the same as the background.

### **STUCK IN TYPING MODE**

Press CTRL-O (Options) and select Option 4.

### **WHICH PAGE AM I LOOKING AT?**

Usually Page 1. The only time you see Page 2 is through Option 6-P or M. If the screen notes are visible or if you have a cursor, you are looking at Page 1.

### **HOW DO I MAKE A PRINTOUT OF MY PICTURE?**

You need two things-- a printer capable of printing hi-res AND hi-res "dump" software. Ask your Apple dealer.

### **CAN I USE DIFFERENT TYPEFACES WITH ALPHA PLOT?**

No. You can, however, add different type styles to your Alpha Plot pictures using other software (Beagle Bros' APPLE MECHANIC disk, for example).

## Also on the Alpha Plot Disk

### ALPHA BOOT

This is the program that runs when you boot. It simply BLOADs Alpha Code, Alpha Plot's machine language programs, sets the Start of Program pointers to location 24577 (\$6001), then RUNs Alpha Plot. If you run Alpha Plot and Alpha Boot hasn't been run, Alpha Plot will run Alpha Boot for you.

### ALPHA CODE

This is the machine language section of Alpha Plot that does Page switches, superimposing, and so on. It is BLOADed by Alpha Boot.

### ALPHA INFO

Run this program to see the latest changes (if any) since these instructions were printed.

### HI-LO PLOT

This program is fun if you like fooling around with graphics. The program actually moves Hi-Res image values from Page 1 (\$2000) or Page 2 (\$4000) down to LO-RES Page 1 (\$400). What you get is an abstract image of your Hi-Res picture in Lo-Res. Since the Hi-Res screen is much bigger than the Lo-Res screen, several versions of the picture are available. Follow the screen instructions. But first, do this--

1. Run Hi-Lo Plot.
2. Quit by pressing Q.
3. Load a Hi-Res picture onto Page 1 by typing "HGR" and "BLOAD FILENAME,A\$2000". A good one to use is the "BBROS LOGO" on the Alpha Plot disk. Load a second picture onto Page 2 if you want by typing "HGR2" and "BLOAD FILENAME,A\$4000".
4. Type "RUN", and select Option H. Follow the instructions and let things happen.
5. When you get a picture you like, exit the program with a Q, and type "BSAVE LO-RES,A\$400,L\$400".

Now let's convert the picture BACK TO HI-RES. This will be an abstract of an abstract, not a real conversion.

6. Load the Lo-Res picture you just made onto LO-RES PAGE 2 by typing "BLOAD LO-RES,A\$800". We have to use Page 2 because Apple's Text shares Page 1 with Lo-Res.
7. Type "RUN" again.
8. Select Option L. There's your abstract of an abstract! To save this picture, type "BSAVE HI-RES,A\$2000,L\$2000".

Keep experimenting. Sooner or later, you'll come up with some award-winners. The actual Hi-Lo Plot program is named "^^". The program you see in the catalog simply changes some pointers to and runs "^^" for you. Don't tell anyone, o.k.?

**SCRUNCH**

Scrunch is a handy machine language program that lets you store hi-res pictures in much less than the normal 34 sectors of disk space, thus postponing DISK FULL error messages. Scrunch is easy to use, especially if you follow these directions carefully--

**TO SCRUNCH (COMPRESS) A PICTURE--**

1. Exit Alpha Plot if necessary. Scrunch has nothing to do with Alpha Plot.
2. Type "BLOAD SCRUNCH" (return).
3. Load a normal 34 sector hi-res picture onto PAGE ONE by typing "BLOAD PICTURE,A\$20000" (return).
4. Type "CALL 25000" (return). This will put a compressed version of your picture onto Page Two. It won't be legible until you unscrunch it (see below).
5. Notice the "BSAVE" message on the screen. Store the scrunched version of your picture (now hiding on Page Two) by typing "BSAVE SCRUNCHPIC,A\$40000,L\$\_\_\_\_" (return). Use any file name you want and the numbers you see on the screen.
6. Your picture is now compressed and stored on disk. Catalog and notice its new size. You may delete the original 34 sector version of the picture AFTER you are sure the compressed version will unscrunch.

**TO UNSCRUNCH A PICTURE--**

1. Type "BLOAD SCRUNCH" (return).
2. Load a scrunched picture onto PAGE TWO by typing "BLOAD SCRUNCHPIC,A\$40000" (return).
3. Clear and view Page One by typing "HGR" (return). This step is optional.
4. Type "CALL 25003" (return).

Scrunch and unscrunch commands may be typed directly from the keyboard or executed from your Applesoft programs. You will only need to bload the Scrunch program once (NOT every time you manipulate a picture). Remember, to use DOS commands like BLOAD and BSAVE under program control they must be preceded by a ctrl-D. Consult your DOS Manual.

The Scrunch program is not relocatable.

The amount of scrunching that takes place on a given image depends mostly on the complexity of that image. Scrunched simple images occupy the least space.



© Bert Kersey, BEAGLE BROS., 4315 Sierra Vista, San Diego, CA 92103  
Telephone (714) 296-6400