

UNPROTECTED

Backups may be made using standard copying procedures.

COMPATIBLE

Standard Apple IIc®  
128K Apple IIe  
ProDOS™ and DOS 3.3



# • BEAGLE GRAPHICS •

16-COLOR DOUBLE HI-RES GRAPHICS  
by MARK SIMONSEN

**DOUBLE HI-RES!** 16 hi-res colors and a 560x192-pixel screen—*twice* the resolution of normal hi-res. All standard Applesoft functions, including shape tables, are supported. All you need is the enclosed disk and an Apple IIc or 128K Apple IIe.

**FOR THE ARTIST:** Draw on the screen using 16 different double hi-res "paintbrushes". **Hi-res Icons** make drawing easy, whether you're using the keyboard or Mouse, Joystick or Graphics Tablet.

**FILL shapes** fast, selecting from 16 solid colors or over 200 hi-res color mixes. Move, invert, or flop any image portion. **Add type to pictures** in different typetypes. Re-define any character.

**FOR THE PROGRAMMER:** Enhance your Applesoft programs with **new commands** that draw *fast* circles and shapes. *Save* any screen area to disk. "Pack" double hi-res images to save disk space.

**"CLIP AND PASTE":** Speeds up screen layouts by letting you move or duplicate any section of a picture anywhere on any picture. *Fast!*

**HI-RES CONVERTERS:** Convert your existing normal hi-res pictures & programs (even *Apple Mechanic* shape table programs) to double hi-res. Just save the new version, and Run to create double hi-res graphics.

**HI-RES PRESENTATIONS:** Organize your double hi-res pictures into an "Apple slide show". Beagle Graphics makes it easy.

**HI-RES HELP:** Use the enclosed (copyable) screen-plotting grid to plot and plan your pictures and charts. A full-color "Help Card" and chart displays all double hi-res colors, commands and procedures.

**FREE PEEKS & POKES CHART:** Apple's "Peeks, Pokes, Pointers & Calls" on one big poster. An indispensable Apple® programming tool.



# • BEAGLE GRAPHICS •

**16-COLOR DOUBLE HI-RES GRAPHICS**  
by **MARK SIMONSEN**

**ISBN 0-917085-02-7**

Instruction Manual  
by **Rob Renstrom & Bert Kersey**

Copyright © 1984, Mark Simonsen  
Published by **BEAGLE BROS, INC.**  
3990 Old Town Avenue  
San Diego, California 92110

## **TABLE OF CONTENTS**

<b>INTRODUCTION</b> .....	3
<b>HARDWARE REQUIREMENTS</b> .....	6
<b>DOUBLE PLOT</b> double hi-res drawing program . . .	11
<b>FOR THE PROGRAMMER</b> .....	31
<b>DHGR</b> double hi-res programming commands . . .	34
<b>DGR</b> double lo-res programming commands . . . . .	64
<b>FONT EDITOR</b> hi-res character changer . . . . .	74
<b>SLIDE SHOW</b> presentation program . . . . .	76
<b>UTILITIES</b> for double-res graphics . . . . .	79
<b>APPENDIX</b> protecting the hi-res screens . . . . .	96
<b>INDEX</b> .....	98



# Introduction



Welcome to BEAGLE GRAPHICS—a complete Double Resolution graphics package for your Apple®. Beagle Graphics' double hi-res feature lets you access a whopping 560 x 192 pixels—that's over *100,000 individual points* on the screen. It also provides you with sixteen undistorted hi-res colors, instead of the six available in normal hi-res. Now all of your Apple graphics—pictures, charts, graphs, even those hi-res portraits of Aunt Sophie—can have more detail and more color!

## **FOR THE ARTIST: DOUBLE PLOT**

The Beagle Graphics disk features a double hi-res drawing program called “Double Plot”, that lets you create colorful shapes, draw points and lines with 16 unique “paintbrushes”, fill shapes in solid or mixed colors, edit pictures with a “cut & paste” feature, add text to pictures... Hey, you're going to *like* Double Plot!

## **FOR THE PROGRAMMER: NEW COMMANDS**

Beagle Graphics provides you with a complete set of double-res Apple graphics commands and functions that can be used in your Applesoft programs. You get double hi-res equivalents of all the standard hi-res commands like HCOLOR, HPLOT and DRAW, plus new commands such as CIRCLE, BOX, FILL, hi-res PRINT, and many more (25 total). We've even added eight double *low-resolution* commands, which let you program in 80 x 48 double lo-res graphics.

## **PLUS...**

There's more— A double hi-res “slide show” program, a hi-res font editor, and some nifty utility programs that let you do some truly amazing graphics tricks—instantly change any color to any other, convert regular hi-res pictures and programs to double hi-res, save and load any portion of a double hi-res picture... Enough to keep you off the streets for weeks!

## About The Disk

First of all, the Beagle Graphics disk is *stuffed*, so any saving of pictures or programs will have to be done on your own disks, formatted (initialized) in the appropriate DOS. Since you're dealing with standard unprotected disks, you can transfer files from disk to disk with Apple's FID (DOS 3.3) or FILER (ProDOS) programs.

### FORMATTING BLANK DISKS

**DOS 3.3:** Use the INIT command (see your Apple DOS 3.3 manuals).

**ProDOS:** BRUN the FILER program from the ProDOS disk (comes with newer Apples).

### DOS 3.3 AND ProDOS™

Two complete sets of programs are included in your Beagle Graphics package (either as two disks or on two sides of the same disk)—a "ProDOS" version and a "DOS 3.3" version. *Both versions work identically.* The one you use is up to you; it depends on which DOS (Disk Operating System) you like to use. You will probably always want to use the same version of Beagle Graphics, so hide the other one now and you won't get confused.

In a way, it doesn't matter which version you use, because pictures saved on disk may be converted from 3.3 to ProDOS and back using ProDOS's "CONVERT" program (don't try to convert the Beagle Graphics *programs*, though). If you choose to use the DOS 3.3 version of Beagle Graphics, we highly recommend using Beagle Bros' **ProntoDOS** utility to *triple the speed* of loading and saving pictures.

### BACK IT UP

In keeping with Beagle Bros tradition, the Beagle Graphics disk is unlocked and unprotected. This means that you can copy it, catalog it, list and modify the programs, experiment with it, and, by the way, *ruin* it— so make a back-up copy now! But, please, *please*, don't give copies away to your friends.

YOU SUPPORT US AND WE'LL SUPPORT YOU.

# Beagle Graphics Disk Catalog

Here is a rundown of most of the files on the Beagle Graphics disk. Type "CATALOG" to check out your disk.

**STARTUP:** *The Beagle Graphics boot-up "greeting" program*

**DHGR:** *BRUN DHGR to load double hi-res graphics commands.*

**DGR:** *BRUN DGR to load double lo-res graphics commands.*

**DOUBLE.PLOT:** *Double hi-res graphics drawing/typing program*

**DP.OBJ1** and **DP.OBJ2:** *Double Plot's machine-language code*

**DP.BAS:** *Double Plot's Applesoft code*

**MOUSE.DRIVER, etc.:** *Double Plot's pointing device code*

**PATTERNS & PATTERNS.AUX:** *Double Plot's color mix picture*

**NOTES:** *Type "RUN NOTES" to read about any changes that might have been made since this manual was printed.*

## **DOUBLE HI-RES UTILITIES:**

**SLIDE.SHOW:** *Double hi-res picture-display program*

**FONT.EDITOR:** *Double hi-res character editor*

**HGR.TO.DHGR:** *Converts hi-res programs to double hi-res*

**CONVERT.HIRES.1:** *Converts pictures to 1/2-size double hi-res*

**CONVERT.HIRES.2:** *Converts pictures to full-screen double hi-res*

**CHANGE.COLORS:** *Changes any double hi-res color to any other*

**DOUBLE.SCRUNCH:** *Compacts pictures to save disk space*

**CUT.AND.PASTE:** *Lets you load, save & move picture sections*

**PAGE.2:** *Lets you move and swap pictures between pages 1 and 2*

## **DOUBLE LO-RES UTILITIES:**

**GR.TO.DGR:** *Converts lo-res programs to double lo-res*

**CONVERT.LORES.1:** *Converts pictures to 1/2-size double lo-res*

**CONVERT.LORES.2:** *Converts pictures to full-screen double lo-res*

**LORES.LOAD.SAVE:** *Lets you load and save double lo-res pictures or 80-column text screens*

## **DOUBLE HI-RES PICTURES:**

*(RUN SLIDE.SHOW to view all of the pictures on the disk.)*

**BEAGLE.PAC, BBROS.PAC, etc.:** *"Packed" pictures.*

## **CHARACTER SETS:**

**ASCII.FONT, BLIPPO.FONT, etc.**

# Hardware Requirements

To use Beagle Graphics, you will need an *Apple IIc* or a 128K *Apple IIe*. Apple II's and II+'s can't handle it. If any new version of the Apple II has been released since the IIc, Beagle Graphics will undoubtedly work on it too.

64K Apple IIe's may be upgraded to 128K (more memory) by adding an Apple-brand EXTENDED 80-COLUMN CARD. A few early-1983 Apple IIe's will need to have a "Revision B" motherboard installed to access double hi-res. Look inside your Apple IIe, somewhere behind slot 4; the number on the green floor should read something like "820-0064-B" (not "-A"). If Beagle Graphics doesn't work on your IIe, this may be your problem; see your Apple dealer for a quick fix.

Note: Your Extended 80-Column Card must have its "jumper" installed. See the "Double High-Resolution Graphics" section of your *Extended 80-Column Text Card Supplement*.

## Monitors

Any type of monitor or TV that will connect to your Apple will work with Beagle Graphics. You will, however, obtain strikingly different results with different monitors.

**MONOCHROME MONITORS** (black & white, green, amber, etc.): You will (obviously) not see colors displayed on the screen. Instead, each color will appear as a different pattern or shade of "grey". 80-column text will be sharp and easy-to-read.

**COLOR TV's and COMPOSITE COLOR MONITORS:** Colors will be true, but will tend to blur or "smear". In the 560 Mode (see next page), supposedly black and white pixels will sometimes appear in color. You may also have a problem reading 80-column text-screen text (40-column text isn't available with double-res pictures). The 140 Modes' wide graphics-screen text, however, will be easy to read, in any color.

**RGB MONITORS:** These big-bucks guys give you razor sharp graphics, stunning colors, and easy to read 80-column text. You will get sixteen colors in the 140 Mode and black and white in the 560 Mode.

## Double Hi-Res Modes

There are four different viewing and plotting "modes" available when using double hi-res graphics:

1. **560 MODE**
2. **140 MODE**
3. **560 MIXED MODE**
4. **140 MIXED MODE**

Each mode can be selected from your programs (page 50) or from Double Plot (page 24). **You will learn far more by experimenting** with your particular monitor than you will by reading the following descriptions:

### Non-RGB Monitors Only

Only 560 Mode (1) & 140 Mode (2) are significant on non-RGB monitors. However, programs that may later be viewed on RGB monitors should follow the rules for all four modes.

**560 MODE:** 560 x 192 resolution with 16 colors. Color lines will sometimes appear broken. Text printed from this mode will be skinny (80-columns), readable in black and white only.

**140 MODE:** 140 x 192 resolution with 16 colors. Color lines will be solid and thick (4 pixels wide). Text printed from this mode will be wide (20-columns), readable in any color.

### RGB Monitors Only

Certain screen images viewed on a RGB monitor will look entirely different in each of the four modes.

**560 MODE:** Same as non-RGB, except **no color** will be visible, even though you may use color commands. Each pixel is either ON (white) or OFF (black).

**140 MODE:** Same as Non-RGB 140 Mode (above).

**560 MIXED MODE:** A combination of 560 and 140 Modes. If used properly,\* you can display true 560-pixel resolution along with 16 color, 140-pixel resolution. After activating 560 Mixed Mode, subsequent drawing will be done as if you were in 560 Mode (the screen will be 560 pixels wide).

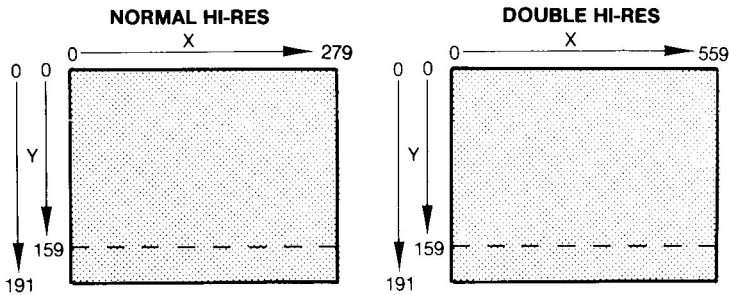
**140 MIXED MODE:** The same as 560 Mixed Mode, except that subsequent drawing will be done as if you were in 140 Mode (the screen will be 140 pixels wide).

\* The general rule is, with some exceptions, don't let anything drawn in the 560 Mode touch anything drawn in the 140 Mode. Programmers Note: 560 Mode pixels and 140 Mode pixels cannot occupy the same screen byte.



## The "Res" Report

Since the beginning of time (1977), Apples have supported "low resolution" graphics and "high resolution" graphics. "Lo-res" has 16 colors and a 40-dot (horizontal) x 48-dot (vertical) screen; its graphics look a lot like the old "Pong" games. "Hi-res" has more detail and less color than lo-res; it lets you plot on a 280-dot (horizontal) x 192-dot (vertical) screen in six colors (plus an extra white and extra black).



### **DOUBLE-RES: EQUAL Y AND DOUBLE X**

Thanks to fancier hardware and more memory per Apple, software like Beagle Graphics can now help double the number of horizontal lo-res and hi-res plots, making each "pixel" (picture element) half its original width. And the number of double hi-res colors is increased to sixteen, like lo-res.

### **DISADVANTAGES? A COUPLE...**

Nobody's perfect. Double-res pictures take twice as long to load and save as normal pictures (see the *ProntoDOS* plug on page 4). And double hi-res pictures occupy two file name positions in catalogs. And twice the disk space. Other than that, no disadvantages.

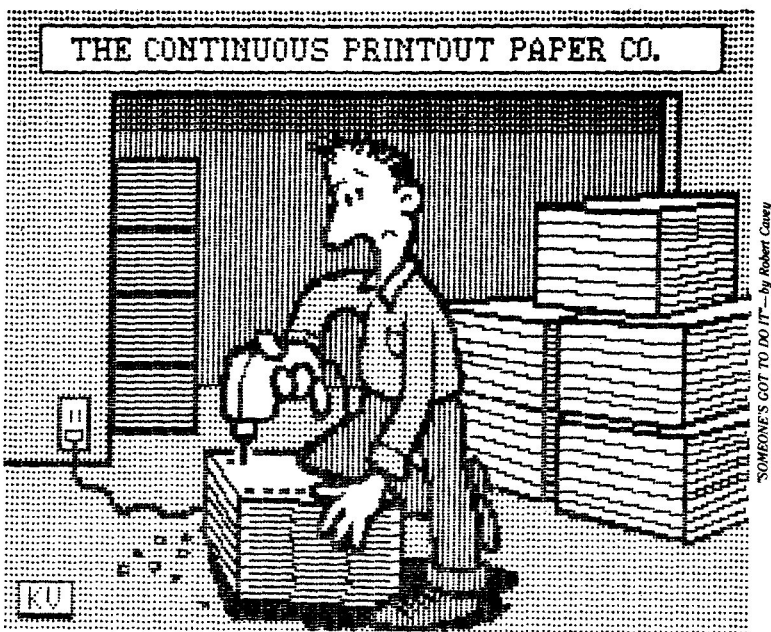
**Use the DOUBLE.SCRUNCH utility** (page 86) to reduce the amount of data used by each double hi-res picture, and to limit each one to only one file name.

## Printing Double-Res Pictures

There's one way we know of—Run out and buy Beagle Bros' *Triple-Dump* "Print Anything" disk. As long as you have a dot-matrix printer capable of printing graphics, *Triple-Dump* will send it all kinds of Apple images:

1. Hi-Res & Double Hi-Res
2. Lo-res & Double Lo-Res
3. 40 & 80-Column text

Actually, *anything* you see on your screen can be printed on paper; cropped, rotated, inversed, stretched, and so on. All of the routines are transferable to your Applesoft programs.





## **New Feature!**

Run the NOTES program to  
see how to use the new  
*Fatbits/Zoom* feature of  
Beagle Graphics.

# DOUBLE.PLOT



## FOR THE ARTIST

With all of the preliminaries out of the way (you did *read* everything, didn't you?), we are ready to do some drawing. No programming in this section, just good old-fashioned (?) electronic drawing with Beagle Graphics' DOUBLE PLOT program.

**Here's what you do to get started:** First, decide if you want to use the ProDOS or DOS 3.3 version of Beagle Graphics (see page 4). Insert the corresponding Beagle Graphics disk, into your disk drive (don't forget to close the door... no, the *disk drive* door!).

Now press **Left-Apple** (*Open-Apple*) **Control-Reset** (all three keys at the same time). This will boot the disk and get you going.

**OR**, from Applesoft, you can simply type (in *UPPER* case, please): **RUN STARTUP**

Either way, you will soon be presented with the Beagle Graphics Opening Menu:

- D. Run DOUBLE.PLOT**
- F. Run FONT.EDITOR** (see page 74)
- S. Run SLIDE.SHOW** (see page 76)
- H. Load DHGR** (see page 34)
- L Load DGR** (see page 64)
- Q. Quit**

Note: Rumor has it that there's a "bug" in this menu if you let it sit around too long without making a selection. WE'VE never seen it though...

Choose option **D** to run the Double Plot Program. You will soon see the Pointing Device Selection Menu:

- M. AppleMouse //**
- G. Apple Graphics Tablet**
- J. Joystick, Paddles, Koala Pad...**
- K. Keyboard**

Choose your pointing device by selecting the appropriate letter. If you have no devices connected to your Apple, choose option **K**, Keyboard. Read more about pointing devices on the next two pages.

To choose a new pointing device, quit and type "**RUN DOUBLE.PLOT**" (Important: see [Q] Quit on page 25).

## Double Plot Pointing Devices

When you use DOUBLE PLOT (Beagle Graphics' drawing program), you will be asked to identify the type of "pointing device" that you want to use—mouse, graphics tablet, joystick, etc., or keyboard. All of these devices perform in nearly the same manner. There are basically two operations which you may perform with each pointing device:

1. **Move something** around on the screen.
2. **Press "The Button"** to start or stop a function. Whenever you see a reference to "The Button" in this manual, it will depend on which pointing device you are using.

### MOUSE

Mice are nice—easy to control and friendly to use. Reasonably priced too. More and more Apple software will be supporting the mouse. Beagle Graphics supports the *AppleMouse* //.

**To Move:** Move the mouse across your desk.

**The Button:** There's only one, so press it!

### GRAPHICS TABLET

The Apple Graphics Tablet provides the easiest, most natural way of drawing and tracing images. They run about 800 bucks apiece; run out and buy a few.

**To Move:** Move the pen lightly across the tablet.

**The Button:** Press the **pen-tip** down (gently) against the Graphics Tablet surface.

## **JOYSTICK, PADDLES, KOALA PAD, etc.**

The Koala Pad™ is a mini-graphics tablet; nice and reasonably-priced, but not as accurate (large) as Apple's. Paddles are difficult to use because two knobs are involved. Joysticks are a better choice.

**To Move:** Move the joystick stick, paddle knob(s), or Koala Pad drawing tool.

**The Button:** Use **BUTTON #0** (usually the left button). Keep reading to learn about the other button.

**Joystick Move-Modes:** You will notice that a joystick (or paddles, Koala Pad, etc.), will move the cursor about three pixels at a time. This "Quick Mode" compensates for the fact that the joystick has only 256 different positions, while the double hi-res screen has 560. To move one pixel at a time, press **BUTTON #1** (usually the right button) to select the "Precision Mode", indicated by a clicking sound. Pressing **BUTTON #1** again will return you to the Quick (and silent) Mode.

## **KEYBOARD**

One nice thing about the keyboard is that every Apple has one. Beagle Graphics works just fine with the keyboard as the pointing device.

**To Move:** Press one of the four arrow keys.

**The Button:** Press the **LEFT-APPLE** key (to the left of the space bar; this key is often called "*Open-Apple*", but we like "Left"). Keep reading to learn about the other APPLE key.

**Keyboard Move-Modes:** You will notice that the arrow keys move the cursor more than one pixel at a time. This "Quick Mode" allows you to move quickly around the screen (at one-pixel per move, it would take forever, or longer). Press the **RIGHT-APPLE** ("Closed-Apple") key to select the "Precision Mode", indicated by a clicking sound. Pressing the **RIGHT-APPLE** key again will return you to the Quick (and silent) Mode.

## The Main Menu

Now you will see the Double Plot main menu below the graphics screen. The main menu shows the various command options (BOX, EDIT, etc.). The remainder of the screen is the “drawing surface” for the double hi-res screen. The main menu is not part of the picture; it will disappear when you begin drawing:

[B] BOX	[E] EDIT	[M] MODE
[C] CIRCLE	[F] FILL	[S] SET COLOR
[D] DRAW	[P] PAINT	[X] CLEAR SCREEN
[L] LINE	[T] TEXT	[Q] QUIT

**Note:** If you are using a monitor that produces fuzzy hard-to-read text, refer to the commands on the color Help card that came with your Beagle Graphics disk.

Now let's experiment with the Double Plot commands described on the following pages. To select a command, simply type its letter (**B** for Box, **E** for Edit, and so on).

### **ABOUT [ESC] AND [RETURN]**

Pressing the [ESC] key will almost always let you “escape” from any situation, whether you have started to mess up the screen, or if you're just plain lost. Pressing the [RETURN] key from the main menu will always “return” you back to the last function used. If you aren't sure what the last function was, look at the upper-right portion of the main menu; you will see “Return=” followed by the last command used.

## [B] BOX

Press **B** from the main menu to draw fast boxes on the screen. You will be presented with a crosshair-cursor which may be moved with your pointing device (mouse, arrow keys, joystick, etc.).

**To draw a box on the screen**, position the crosshair where you want one corner of the box. Now press The Button\* and continue holding it down while you move the cursor around the screen. You will see an elastic, or "rubber band", box that will grow or shrink according to the crosshair's position. Once you see a box you like, release The Button and the box will be printed on the screen. The crosshair may now be moved around the screen to make more boxes. Fun, huh?

**If you begin to make a box and change your mind**, just press the [ESC] key *before releasing The Button*, and the box will disappear. The crosshair will re-appear as soon as you release The Button.

**Note:** A perfect square on one monitor may have non-equal sides on another. Your monitor might have some kind of adjustment for this (and it might not).

**To exit the Box function** and return to the main menu, press [ESC] when you are not pressing The Button.

## [C] CIRCLE

Pressing **C** from the main menu allows you to draw circles and ellipses in the same manner you draw boxes (above). Try it and you'll see what we mean.

**If you start an ellipse and change your mind**, press the [ESC] key before releasing The Button.

**Note:** A perfect circle on one monitor may look like an egg on another. See if you can adjust your monitor.

**To exit the Circle function**, press the [ESC] key when you are not pressing The Button.

\* Depends on which pointing device you have selected. Press the Mouse Button or Graphics Tablet Pen or Joystick/Paddle Button #0 or Keyboard Left-Apple key.



## [D] DRAW

Pressing **D** from the main menu allows you to do “free-hand” drawing on the hi-res screen. You will be presented with a hi-res pencil as a cursor. This pencil may be moved around the screen with your pointing device.

**To draw freehand lines**, press The Button.\* Hold The Button down while you move the pencil with your pointing device. To draw a single point, press The Button once and release it.

**The only way to erase freehand lines** is to draw over them in the background color. OR load (page 27) the earlier version of your picture that you remembered to save on disk. You did remember, didn't you?

**To return to the main menu, press [ESC].**

## [L] LINE

Typing **L** from the main menu lets you draw straight lines (as opposed to option D's freehand lines). You will be presented with a movable hi-res pencil as a cursor.

**To draw a straight line**, move the pencil to the starting point of the line. Now press The Button\* and continue holding it down while moving the pencil. You will see an elastic line that changes as you move around. Once you have the desired line, release The Button and the line will be made permanent.

**If you start to make a line and change your mind**, press the [ESC] key before releasing The Button, and the line will vanish.

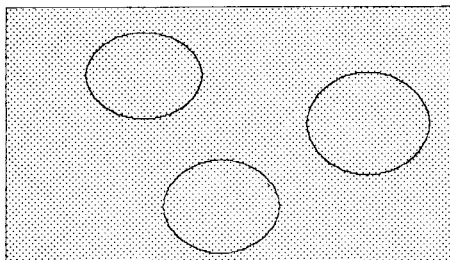
**To return to the main menu, press [ESC].**

\* Depends on which pointing device you have selected. Press the Mouse Button or Graphics Tablet Pen or Joystick/Paddle Button #0 or Keyboard Left-Apple key.

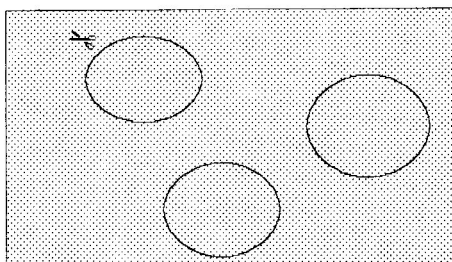
## [E] EDIT

Press **E** from the main menu to select Edit Mode, one of Double Plot's most powerful features. It allows you to perform a graphics "Cut and Paste" operation—"cutting" an area from the double hi-res screen and then "pasting" it back at a different position. You can even paste it onto a different picture, as many times and places as you want.

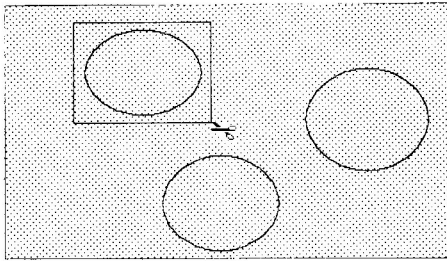
To demonstrate cutting and pasting, we first need a picture to work with. Let's say we have some circles:



Now, let's move one of the circles. Press **E** from the main menu. You will be given a pair of scissors as your cursor (indicating the CUT MODE). Now select an area to edit (move) by moving the scissors to the upper left of one of the circles:



Now hold down The Button\* and move the scissors diagonally to the lower right of the circle:



The elastic box connected to the scissors indicates the area to be edited. Release The Button and you will be placed in the **PASTE MODE**. At this point, you have the following options:

**[ESC]** exits Paste Mode.

**I** inverts the selected area (neg. image)

**H** horizontally flips the area.

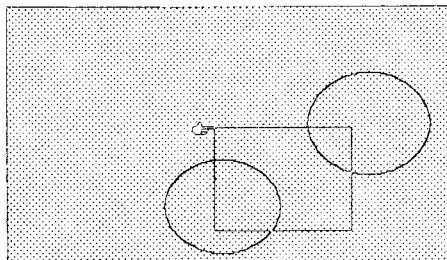
**V** vertically flops the area.

} See  
next  
page

**M** moves the area into memory, saving it for pasting. The area will be erased in the background color (see "Set Color" on page 24).

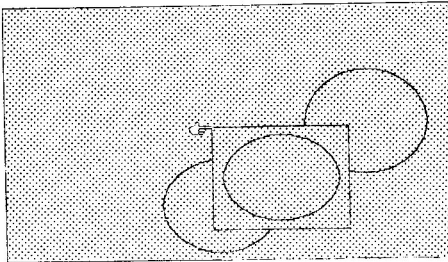
**C** copies the area into memory, saving it for pasting (just like Move, but the original area won't be erased).

To move the circle in our example, type **M** (for "Move"). The area within the box will be erased and stored in memory. You will now see a pointing-finger cursor and a box indicating the size of the area that was "cut out":



\* Depends on which pointing device was selected. Press Mouse Button or Graphics Tablet Pen or Joystick/Paddle Button #0 or Keyboard Left-Apple key.

By moving the pointing finger around the screen, you can “paste” a copy of the selected area anywhere by pressing The Button. You can make as many copies as you want by pressing The Button some more.

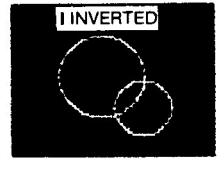
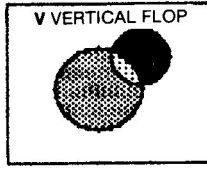
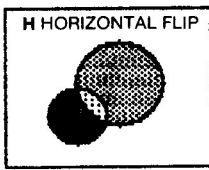
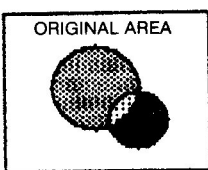


To return to the Cut Mode (and the scissors cursor) press the [ESC] key. Pressing [ESC] a second time will return you to the main menu.

### PASTING ONTO ANOTHER PICTURE

1. Define the area with the Cut Mode's scissors.
2. Store the desired area in memory using the Cut Mode's **C**opy or **M**ove options.
3. Return to the main menu and load or create another picture (using *any* command *except* **E**dit).
4. From the main menu, type **Left-Apple E** (both keys at the same time). This will place you directly into the Paste Mode (skipping the Cut Mode).
5. You can now paste the stored area anywhere on the screen, any number of times, by pressing The Button.

Here are some examples of the other Paste Mode options. Press **I**, **H** or **V** after defining a rectangular area with the Cut Mode's scissors:



### ERASING AN AREA

To quickly erase a rectangular area of the screen—just “cut out” the area and don't bother to paste it back!

## [F] FILL

Typing **F** from the main menu allows you to fill an area with one of 256 fill colors and patterns.

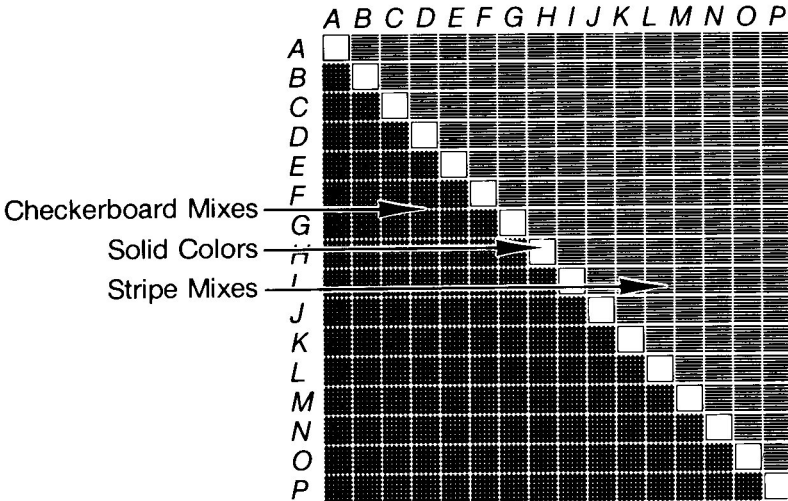
**IMPORTANT:** Only two kinds of areas can be filled with Beagle Graphics' Fill routine: BLACK AREAS surrounded by a WHITE BORDER (or the edge of the screen), and WHITE AREAS surrounded by a BLACK BORDER (or the edge of the screen). If you try filling a color area or a black or white area surrounded by a color border, you're taking your chances. Go ahead and experiment; it's your Apple.

Sixteen of the fill colors are solid colors. The remaining 240 are *mixes* of the 16 double hi-res colors: 120 are "checkerboard" (every other *dot*) combinations, and 120 are "stripe" (every other *line*) combinations.

### FILL PATTERN SAMPLER

After pressing **F** from the main menu, you can see samples of all 256 possible fill colors and patterns by typing **control-F**:

This picture should help you sort out the color mixes:



Press any key to return to the Fill color selection menu.

You will be asked to select the FIRST and SECOND fill colors. This gives you the option of mixing colors. Press [ESC] if you don't want to make a choice.

**FOR SOLID COLOR FILLS:** Specify the *same* color-letter for both the first and second choices.

**FOR CHECKERBOARD FILLS:** Select the lower-letter color first, and *then* the higher-letter color (for example, color A, then H).

**FOR STRIPE FILLS:** Select the higher-letter color first, and *then* the lower-letter color (for example, color H, then A).

Once the fill color(s) are chosen, you will be presented with an arrow cursor. Move the arrow to the desired location (probably within an enclosed area) and press The Button.\* The area will immediately begin to fill with color. When the fill is completed, the arrow will appear again, allowing you to fill another area. If you see that the fill isn't doing what you want, press the [ESC] key and it will stop immediately. Any corrections will have to be done "by hand", so keep your finger near the [ESC] key when doing fills.

**Play it safe:** Save valuable pictures before filling. If something goes wrong (pictures occasionally have unnoticed "leaks"), simply load the original back onto the screen.

**Note:** To skip the color-selection step, type **Left-Apple F** from the main menu. This will place you directly in Fill Mode with the most recent choice of fill colors.

**Pressing [ESC] will return you to the main menu.**

\* Depends on which pointing device was selected. Press Mouse Button or Graphics Tablet Pen or Joystick/Paddle Button #0 or Keyboard Left-Apple key.

# [P] PAINT

When **P** is typed from the main menu, you will be asked to select one of sixteen double hi-res "paintbrushes". Each one produces a different brushstroke:



Enter your brush selection (A-P), or press the [RETURN] key to select the current brush (indicated by the arrow below the brush letters). To paint, hold down The Button and move the cursor with your pointing device. Release The Button to stop painting.

You will notice that moving the brush quickly will produce a scattered brushstroke. Moving it slowly makes a solid stroke (just like a real paintbrush). Experiment to get the feel of each brush.

Painting in the background color is one good way to erase parts of pictures. (Using the Edit function is another.)

**Note:** To enter Paint mode using the most-recently selected brush, type **Left-Apple P** from the main menu.

**To exit the Paint function, press the [ESC] key.**

\* Depends on which pointing device was selected. Press Mouse Button or Graphics Tablet Pen or Joystick/Paddle Button #0 or Keyboard Left-Apple key.

## [T] TEXT

Pressing **T** from the main menu allows you to add text to your double hi-res pictures. You will be presented with a movable "I-Bar" for text positioning. Move the bar to any position on the screen, then press The Button.\* You may now type text from that position.

While typing, you can move the text cursor (an underscore) around the screen with these keys:

**[RETURN]** moves to the first column of the next line.

**control-W** moves up one line.

**control-Z** moves down one line.

**control-A** moves left one character.

**control-S** moves right one character.

(Notice the diamond layout of W, Z, A and S.)

**Character colors** may be changed with the [S] Set Color command (next page). The foreground color is the color of the character. The background color is the color "under" the character.

**Large characters** (*wide*, that is) may be created by typing while in Modes 2 or 4 (next page). Taller-than-normal text is not available with Beagle Graphics.

**Different typefaces** may be loaded from the Beagle Graphics disk or from Apple's *DOS Tool Kit*<sup>™</sup> disk (see "Font" on page 26). *Apple Mechanic's* fonts will not work with Double Plot or Beagle Graphics; sorry. You *may* use *Apple Mechanic* to add text to double hi-res pictures after converting its programs so they function in double hi-res (see the HGR.TO.DHGR utility on page 88).

**Pressing [ESC] will return you to the main menu.**

\* Depends on which pointing device was selected. Press Mouse Button or Graphics Tablet Pen or Joystick/Paddle Button #0 or Keyboard Left-Apple key.



## [M] MODE

Pressing **M** from the main menu lets you change the double hi-res modes (see page 7):

**1: 560 MODE:** 560 x 192 black and white (if RGB).

**2: 140 MODE:** 140 x 192 with sixteen colors.

**3: 560 MIXED MODE:** A combination of Mode 1 and Mode 2. Drawing will be done as if you were in Mode 1 (560 pixels wide).

**4: 140 MIXED MODE:** A combination of Mode 1 and Mode 2. Drawing will be done as if you were in Mode 2 (140 pixels wide).

The best way to understand Apple's modes is to experiment. Now's your chance.

**Press [ESC] to exit without making a choice.**

## [S] SET COLOR

Pressing **S** from the main menu allows you to choose a "foreground" color for drawing lines and typing characters, and a "background" color to be used by the Edit, Text, and Clear Screen functions. To make a color choice, type a letter (A-P), or simply press [RETURN] to select the current color, indicated by an arrow under that color's letter.

The letters A-P correspond to color numbers 0-15. We used letters so you could enter a color with one keystroke.

**Press [ESC] to exit without making a choice.**

## [X] CLEAR SCREEN

Pressing **X** from the main menu lets you clear the screen to any color (think twice before using this option). When asked to enter a screen color (A-P) you may simply press [RETURN] to select the current background color, indicated by an arrow under the color's letter.

To erase *part* of an image, use Edit (page 19) or Paint (page 22).

**Press [ESC] if you don't want to clear the screen.**

## [Q] QUIT

We'll leave it to you, as an exercise, to figure this one out. Oh, if you accidentally quit Double Plot, you can save the picture on the screen by typing "& SAVE, etc." (see page 55). Then type "RUN" (with no file name) to re-start Double Plot.

## [control-D] DISK COMMANDS

Ok, you've created a stunning double hi-res masterpiece and want to save it for posterity (or at least long enough to show it off). To do this, you need to communicate with your disk drive. This can be accomplished like so:

Type **control-D** from the main menu to bring up a secondary menu of disk related functions:

**[C] CATALOG**    **[D] DISK DRIVE**  
**[F] FONT**  
**[L] LOAD**  
**[S] SAVE**

This menu will allow you to Save and Load double hi-res pictures to and from disk, Catalog a disk, load a hi-res character Font, or switch between two disk drives, if you have two drives.

### C: CATALOG

(Press **C** after typing **control-D** from the main menu.)

Typing **C** from the disk menu displays the catalog of the disk in your active disk drive. See Load and Save (pages 27-28) for information on recognizing double hi-res pictures in a catalog.

### F: FONT

(Press **F** after typing **control-D** from the main menu.)

Typing **F** from the disk menu allows you to load a hi-res character font to be used with the [T] Text command. We have provided several character fonts on the Beagle Graphics disk (the files that end with ".FONT"). Only one may be in memory at a time.

Type the name of the desired font to load it. Or press [ESC] to exit without making a selection.

## L: LOAD

(Press L after typing **control-D** from the main menu.)

Press **L** from the disk menu to load a picture. Insert a disk with picture files on it, and type the name of the picture to be loaded.

**Type only one file name** (without the “.AUX” suffix) even though two files are actually loaded. Double hi-res pictures are stored as two separate files, one file for “main memory” and another file for “auxiliary memory”.

The Load function assumes that the two names that make up a double hi-res picture are identical except that one ends in “.AUX”. For example, if you enter “DOG” as the picture to be loaded, the Load function expects to find the two files “DOG” and “DOG.AUX” on the same disk. *Rename* the files (see your Apple manual) if they aren't named properly. (The Save function adds “.AUX” automatically.)

An “Unable to Load File” error message probably means the picture you selected isn't on the disk.

Pictures that have been packed with the Beagle Graphics DOUBLE.SCRUNCH utility cannot be loaded from Double Plot. Unpack any packed picture that you want to change and save it using the & SAVE command.

**Press [ESC] to exit without loading a picture.**

## S: SAVE

(Press **S** after typing **control-D** from the main menu.)

Typing **S** from the disk menu lets you save a double hi-res picture on disk. Insert a disk with plenty of space (see Disk Notes below), and type the name of the picture you want saved.

**Important:** Keep your ProDOS picture names **under 12** characters, and your DOS 3.3 names **under 27** characters (leaving room for the four-character “.AUX” suffix). Type *only one name*, even though two will actually be saved. See the notes regarding file names under Load (previous page).

**Disk Notes:** There isn't enough room on the original Beagle Graphics disk to save a double hi-res picture; use a disk with more space. Allow **68 sectors** (DOS 3.3) or **34 blocks** (ProDOS) per picture. You may only save pictures onto a disk that has been initialized (formatted) in the same format as the version of Beagle Graphics that you are using (DOS 3.3 or ProDOS). See your Apple manuals and page 4 of these instructions.

An “**UNABLE TO SAVE FILE**” error message probably means that you haven't specified the correct prefix (ProDOS only).

**Press [ESC] to exit without saving a picture.**

## D: DRIVE

(Press **D** after typing **control-D** from the main menu.)

### DOS 3.3 DRIVE SELECTION

Typing **D** from the disk menu lets you switch between disk drives 1 and 2. This works as a “toggle” function—pressing **D** once will select drive 2. Pressing **D** a second time will switch back to drive 1, etc.

### ProDOS DRIVE SELECTION

Typing **D** from the disk menu lets you type “,D1” or “,D2” for drive 1 or drive 2. Or you may enter a new prefix.

## Double Plot Program Notes

When you first run Double Plot, all of these files must be on the same disk:

DOUBLE.PLOT  
DP.OBJ1 *and* DP.OBJ2  
DP.BAS  
DHGR  
PATTERNS  
PATTERNS.AUX  
MOUSE.DRIVER...  
(or the DRIVER for your pointing device)

After Double Plot has loaded, you may put the disk away and insert your picture disk (or, of course, you may use two drives).

DP.BAS, Double Plot's main Applesoft program, uses the ampersand (&) commands that are described in the following section. You may make all the changes you want to DP.BAS, but *watch out*; it is literally crammed into memory and has very little or no room to grow.

### ICON SELECTION

Double Plot uses the arrow, crosshair, finger, scissors, I-bar, and pencil as cursors. If you want to switch cursors so that, for example, the scissors can be used to draw (you're *weird*, aren't you?), look at Line 1 of DP.BAS:

**1 ARROW=1: PENCIL=2: CROSS=3:  
BAR=4: FINGER=5: SCISSERS=6**

To change a cursor, simply switch the variables around. To make the scissors draw, let the variable PENCIL equal 6. (Notice that the variable SCISSERS is intentionally misspelled to keep Applesoft happy.)

### QUICK/PRECISION MODE CHANGES

The pixel increment for the keyboard and joystick Quick Mode may be changed in lines 300-330 of DOUBLE.PLOT.



## For the Programmer: 33 New Commands

The rest of this manual is for those of you who write your own programs. First we will discuss Beagle Graphics' 25 new **double hi-res commands**, then cover the 8 new **double lo-res commands** and the Font Editor, Slide Show and **double-res utilities**. Most of your questions can be answered by experimenting—type in the examples, make some changes, and see what happens. If you are working with pictures that you want to keep, **save backups** before you experiment. Then you have nothing to lose.

Both beginners and experts qualify, but we *do* assume that you have some experience with Applesoft hi-res commands (HGR, HPLOT, DRAW, etc.). If you don't, you might want to read about them in your *Applesoft BASIC Programmer's Reference Manual*.

Have fun and make some great Apple graphics!



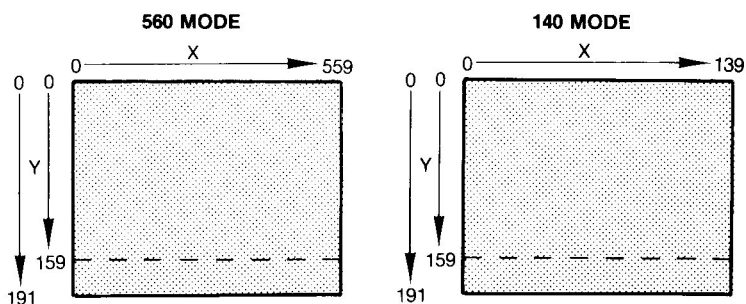
## Screen Layout

A double-res screen layout is included on the back side of the color Help card that came with your Beagle Graphics disk. You may make copies for your personal use.

Many of the new commands use x (horizontal) and y (vertical) screen coordinates to position something on the screen. If you specify coordinates greater than the screen limits, your program will beep to a halt with a well-deserved "?Illegal Quantity Error" message.

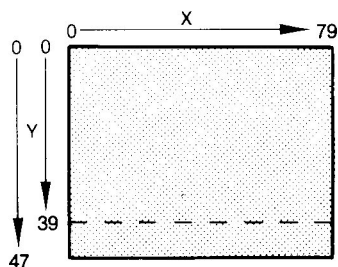
### DOUBLE HI-RES LIMITS

X must be a numeric expression 0-559 (560 Mode) or 0-139 (140 Mode). In both 560 and 140 Modes, Y must be a numeric expression 0-191 (full-screen graphics), or 0-159 (split-screen with four lines of text at the bottom).



### DOUBLE LO-RES LIMITS

X must be a numeric expression 0-79. Y must be a numeric expression 0-47 (full-screen graphics), or 0-39 (split-screen with four lines of text at the bottom).



## Trouble Shooting

If you are writing a double-res graphics program that isn't acting the way you think it should, welcome to the club! Here are a few common problems and solutions. Remember, computers (almost) always do exactly what you tell them.

### **NON-FUNCTIONING COMMANDS**

Perhaps you left the "&" off of a double-res command. Also check and see that you're not plotting in the same color as the background.

### **ZAPPED MEMORY**

You probably forgot to do an & HGR, & HGR2, & GR or & GR2 before you plotted or cleared the screen. ALWAYS begin double-res programs with one of these commands.

### **SPREAD OUT TEXT-SCREEN TEXT**

You probably hit control-reset. Type "PR#3" for a quick fix. You should always be in 80-column mode (not 40-column) when going into double-res.

### **?ILLEGAL QUANTITY ERROR**

You are probably trying to plot beyond the limits of the screen. Check to see that your x values are appropriate for the mode you are plotting from (see page 7). Also remember, boxes and ellipses must fit on the screen in their entirety.

### **HOLES IN DOUBLE HI-RES LINES**

Switch your plotting color to white (15) or plot from 140 Mode (2) or 140 Mixed Mode (4).

### **CAN'T PRINT ON THE GRAPHICS SCREEN**

Perhaps you didn't issue an & NORMAL command the last time you quit printing on the graphics screen. It doesn't hurt to start programs with an & NORMAL.

### **COLOR PROBLEMS**

Maybe you're in 560 Mode and using an RGB monitor. See page 7. Maybe your tv's out of adjustment.

## DHGR (alias Double Hi-res GRaphics)



DHGR is a machine-language program on the Beagle Graphics disk that adds 25 double hi-res “ampersand commands” (special commands preceded by an “&”) to Applesoft BASIC. After loading DHGR, these commands can be used in your Applesoft programs:

<i>New command</i>	<i>Normal hi-res equivalent</i>
<b>&amp; BCOLOR</b>	(none)
<b>&amp; BOX</b>	(none)
<b>&amp; CIRCLE</b>	(none)
<b>&amp; CLEAR*</b>	(none)
<b>&amp; DRAW</b>	<b>DRAW</b>
<b>&amp; FILL</b>	(none)
<b>&amp; GOTO*</b>	(none)
<b>&amp; HCOLOR</b>	<b>HCOLOR</b>
<b>&amp; HGR</b>	<b>HGR</b>
<b>&amp; HGR2*</b>	(none)
<b>&amp; H PLOT</b>	<b>H PLOT</b>
<b>&amp; HSCRN</b>	(none)
<b>&amp; LOAD*</b>	<b>BLOAD</b> a picture
<b>&amp; MODE</b>	(none)
<b>&amp; NORMAL*</b>	(none)
<b>&amp; PRINT*</b>	(none)
<b>&amp; ROT</b>	<b>ROT</b>
<b>&amp; SAVE*</b>	<b>BSAVE</b> a picture
<b>&amp; SCALE</b>	<b>SCALE</b>
<b>&amp; TEXT</b>	<b>TEXT</b>
<b>&amp; XBOX</b>	(none)
<b>&amp; XCIRCLE</b>	(none)
<b>&amp; XDRAW</b>	<b>XDRAW</b>
<b>&amp; XPLOT</b>	(none)
<b>&amp; XPRINT</b>	(none)

\* Command has a different function than its non-& equivalent.

## Loading DHGR

To install DHGR in memory so the new ampersand commands will take effect, type "**BRUN DHGR**" directly from the keyboard, or put this DOS command in an Applesoft program:

**10 PRINT CHR\$(4); "BRUN DHGR"**

That's all there is to it. If you're an advanced programmer, read the rest of this page. Otherwise, you're ready to try some of the double hi-res commands that follow.

### LOADING PROCEDURE

DHGR will load at \$2000 (8192) and then relocate itself in the highest memory available (just below Himem), occupying 4K of memory.

### DOUBLE TROUBLE

Because Beagle Graphics' double-res programs relocate themselves below Himem each time they are Brun, it is possible to waste memory if the program you are loading is already loaded. The easiest way to avoid this problem of "double loading" is to type "**FP**" (DOS 3.3) or "**-FP**" (ProDOS). (A memory-clearing program called "FP" is included on the ProDOS version of Beagle Graphics).

### COMBINING DHGR, DGR AND OTHER & PROGRAMS

DHGR and DGR can be used together or separately, or with other software that uses the ampersand, thanks to *daisy chaining*— When DGR or DHGR is loaded, it looks to see if any other program is using the & vector. If so, & commands not recognized by DHGR or DGR will be sent along to the other routine for processing.

## Command Description Format

Each of the new double hi-res and double lo-res commands will be presented in the following format:

**Syntax:** The correct way to type the command. If you don't understand, try the command examples. The following syntax rules apply:

- Capital letters must be typed as shown.
- Lower case *italic* items are to be supplied by the user (that's you!).
- Items within [square brackets] are optional.
- Punctuation except square brackets must be typed as shown.
- Items followed by an elipsis (...) may be repeated any number of times.

**Example:** A typical way the command would be typed. Some examples assume you have typed other commands first (like & HGR to reveal the double hi-res screen).

**Purpose:** What the command is used for.

**Remarks:** Comments and notes.

### Program Example:

The fun part— The samples assume you have BRUN DHGR (for double hi-res) or BRUN DGR (for double lo-res). Press any key when each example is finished, and the screen will be cleared and the program listed. Try different values and commands, and see what happens. Be sure and type NEW between examples.

The program examples are not crash proof if you enter an illegal value. We figured you'd learn more that way.

## TERMINOLOGY

**Numeric Expression:** A number (like 10), a variable (like B), or a series of numeric operations (like 5+21 or X/2+Y).

**String Expression:** A string literal (like "HELLO"), a string variable (like A\$), or a series of string operations (like X\$+CHR\$(C)+MID\$(A\$,2,3) or A\$+"HELLO").

## & BCOLOR

**Syntax:** & BCOLOR = *color value*

**Examples:** & BCOLOR=15  
& BCOLOR=2

**Purpose:** Sets the background color for clearing the screen (see & CLEAR) and for double hi-res text (see & PRINT).

**Remarks:** *Color value* must be a numeric expression 0-15, specifying a double hi-res color:

0 Black	8 Brown
1 Magenta (Red)	9 Orange
2 Dark Blue	10 Grey 2
3 Violet	11 Pink
4 Dark Green	12 Green
5 Grey 1	13 Yellow
6 Medium Blue	14 Aqua
7 Light Blue	15 White

(See the colors on the Help card that came with your disk.)

In 560 Mode, colors other than zero (black) and 15 (white) may appear as a black and white pattern (see Modes, page 7).

The background color is automatically set to zero (black) whenever & HGR or & HRG2 is executed.

"& BCOLOR" will list as "& B COLOR".

### Program Example:

```
10 & HGR2 : & MODE(2) : HOME : POKE
   - 16301,0
20 FOR C = 0 TO 15
30 VTAB 22: PRINT "THIS IS COLOR
   #";C;"."
40 & B COLOR= C: & CLEAR
50 NEXT C
60 GET A$: HOME : & TEXT : LIST
```

(See & PRINT for an example that uses & BCOLOR with double hi-res text.)

## & BOX

**Syntax:** & BOX (*x length* [, *y length*]) [AT *x,y*]

**Purpose:** Draws a square or rectangle on the double hi-res screen.

**Examples:** & BOX(50,25) AT 80,80  
& BOX(50) AT 70,80  
& BOX(100)

**Remarks:** *X length* specifies the horizontal length of the box. This must be a numeric expression within the screen limits.

*Y length* is optional; it specifies the height of the box. If it isn't specified, then the *x length* will be used for the box's height.

"AT *x,y*" is optional; it specifies the location of the upper-left corner of the box. If it isn't used, then the box will be drawn starting at the last location plotted.

The entire box must fit on screen or you will get an "?Illegal Quantity Error".

The color of the box is determined by the last & HCOLOR command.

**Square Boxes/Circular Ellipses:** (for most monitors) In the 560 Mode, pixels are roughly two times as tall as they are wide. Therefore, if you want to draw a square (instead of a rectangle) or a circle (instead of an ellipse), use a *y length* that is half of the *x length*. In the 140 Mode, pixels are roughly two times as wide as they are tall; to draw a square or circle, use an *x length* that is half the *y length*.

### Program Example:

```
10 & HGR2 : & HCOLOR= 15
30 FOR X = 0 TO 400 STEP 10
50 & BOX(100,50) AT X,X / 4
60 NEXT X
70 GET A$: HOME : & TEXT : LIST
```

## & CIRCLE

**Syntax:** & CIRCLE (*x radius* [, *y radius*]) [AT *x,y*]

**Examples:** & CIRCLE(20,10) AT 100,100  
& CIRCLE(40) AT 70,96  
& CIRCLE(100)

**Purpose:** Draws a circle or ellipse on the double hi-res screen.

**Remarks:** *X radius* is a numeric expression that specifies the horizontal radius of the ellipse.

*Y radius* is optional; it specifies the vertical radius of the ellipse. If it isn't specified, then the *x radius* will be used for the *y radius*.

"AT *x,y*" is an optional parameter that locates the center of the ellipse. If a center isn't specified, then the ellipse will be drawn at the last (*x,y*) location plotted.

The entire ellipse must fit on the screen or you will get an "?Illegal Quantity Error".

The color of the ellipse is determined by the last & HCOLOR command.

See the "Square Boxes/Circular Ellipses" note under & BOX (previous page).

### Program Example:

```
10 & HGR2 : & HCOLOR= 15
20 FOR XR = 200 TO 0 STEP - 10
30 & CIRCLE(XR,90) AT 280,96
40 NEXT XR
50 FOR YR = 90 TO 0 STEP - 10
60 & CIRCLE(200,YR) AT 280,96
70 NEXT YR
80 GET A$: HOME : & TEXT : LIST
```



## & CLEAR

**Syntax:** & CLEAR

**Examples:** & BCOLOR=13: & CLEAR  
& BCOLOR=0: & CLEAR

**Purpose:** Clears the entire double hi-res screen to the color specified by the last & BCOLOR command.

**Remarks:** If an & BCOLOR command hasn't been executed since the last & HGR (or & HGR2), then the screen will be cleared to black.

### Program Example:

```
10 & HGR : & MODE(2)
30 HOME : POKE - 16301,0
40 VTAB 22
50 INPUT "Clear screen to what c
   olor (0-15 or <Return> to ex
   it): ";C$
55 C = VAL (C$)
60 IF LEN (C$) = 0 THEN HOME :
   & TEXT : LIST : END
70 & B COLOR= C: & CLEAR
90 GOTO 30
```

## & DRAW

**Syntax:** & DRAW *shape number* [AT *x,y*]

**Examples:** & DRAW 4 AT 50,100  
& DRAW 3

(assumes a shape table is loaded and pointers are set)

**Purpose:** Draws a shape at a specified location on the double hi-res screen from the shape table currently in memory.

**Remarks:** *Shape number* must be a numeric expression 1-255. This number specifies which shape will be used from the shape table currently in memory.

"AT *x,y*" is optional; it specifies the starting location of the shape. If it isn't used, then the shape will be drawn at the last point plotted.

For more information on shape tables consult the *Applesoft BASIC Programmer's Reference Manual* or see Beagle Bros' *Apple Mechanic* disk and manual. (*Apple Mechanic* lets you draw the shapes and makes your Apple do all the "dirty work" of converting the shapes into data.)

**Notes:** The color, rotation, and scale of the shape must be specified before the & DRAW command is executed.

### Program Example:

```
10 LOC = 24576: REM SHAPE TABLE LOCATION
15 POKE 232,0: POKE 233,96: REM POKE LOC INTO 232-233
20 FOR I = LOC TO LOC + 10: READ A: POKE I,A: NEXT
30 DATA 1,0,4,0,37,53,53,55,39,39,0
40 & SCALE= 9: & ROT= 0
100 & HGR : & MODE(2): & HCOLOR= 2
110 & DRAW 1 AT 50,100
150 GET A$: HOME : & TEXT : LIST
```

## & FILL

**Syntax:** & FILL (*1st color* [,*2nd color*]) [AT *x,y*]

**Examples:** & FILL (2,3) AT 20,34  
& FILL (2)

**Purpose:** Fills an outlined area with the specified color or pattern.

**Remarks:** *1st color* and *2nd color* must be numeric expressions 0-15. See & HCOLOR for color names and values.

If *2nd color* is not specified, then the area will be filled with solid *1st color*.

*There are 240 possible color mixes— 120 Checkerboard mixes (1st color lower than 2nd color), and 120 Stripe mixes (1st color higher than 2nd color).*

"AT *x,y*" is optional; it specifies the start location for the fill. If not used, the fill will begin at the last point plotted. Be sure that *x,y* is within an area that is completely surrounded by a border (or the edge of the screen).

**Panic Button:** If a fill isn't doing what you want, press [ESC] fast, and your program will advance to the next command. Saving pictures before you fill is a wise move.

Only two kinds of areas can be correctly filled—BLACK AREAS completely surrounded by a WHITE BORDER (or the edge of the screen), and WHITE AREAS surrounded by a BLACK border (or the edge of the screen). If you try filling a color area or an area surrounded by a color border, you will obtain unpredictable results.

### Program Example:

```
10 & HGR : & MODE(2) : & B COLOR=
    15: & CLEAR
20 & HCOLOR= 0: & BOX(40,50) AT
    40,100: & BOX(40,50) AT 60,8
    0
30 & GOTO 50,125: & FILL(1,13):
    REM CHECKERBOARD FILL
40 & GOTO 70,90: & FILL(13,1): REM
    STRIPE FILL
50 & GOTO 79,110: & FILL(1): REM
    SOLID FILL
100 GET A$: HOME : & TEXT : LIST
42 Double Hi-Res Commands
```

## & GOTO

**Syntax:** & GOTO x,y

**Examples:** & GOTO 50,75  
& GOTO 99,123

**Purpose:** Positions the invisible double hi-res cursor at the specified (x,y) location. & GOTO is used for positioning text, boxes, circles, shapes, fills, lines, and so on.

**Remarks:** X and y must be numeric expressions within the screen limits.

### Program Examples:

```
10 & HGR2 : & HCOLOR= 15
20 FOR A = 15 TO 180 STEP 30
25 & GOTO A * 3,A
30 & BOX(30,15): & CIRCLE(30,15)
40 NEXT
50 GET A$: HOME : & TEXT : LIST
```

```
20 F = 16384: PRINT CHR$(4);"BL
    OAD ASCII.FONT,A";F
30 L = PEEK(974) + PEEK(975) *
    256: REM DHGR LOCATION
40 POKE L + 3,0: POKE L + 4,64
100 & HGR2 : & PRINT : & HCOLOR=
    15: & MODE(2)
110 & GOTO 0,0: PRINT "+ <-UPPE
    R LEFT"
130 & GOTO 33,184: PRINT "LOWER
    -RIGHT-> +"
140 & MODE(1): & GOTO 290,96: PRINT
    "+ <--MIDDLE"
150 GET A$: HOME : & TEXT : & NORMAL : LIST
```

## & HCOLOR

**Syntax:** & HCOLOR=*color value*

**Examples:** & HCOLOR=9  
& HCOLOR=2

**Purpose:** Sets the display (foreground) color for plotting double hi-res graphics.

**Remarks:** *Color value* must be a numeric expression 0-15, specifying a double hi-res color. This color will be used to draw subsequent points, lines, circles, boxes, shapes, double hi-res text, and so on:

0 Black	8 Brown
1 Magenta (Red)	9 Orange
2 Dark Blue	10 Grey 2
3 Violet	11 Pink
4 Dark Green	12 Green
5 Grey 1	13 Yellow
6 Medium Blue	14 Aqua
7 Light Blue	15 White

(See the colors on the Help card that came with your disk.)

In the 560 Mode, colors other than black and white will appear as a pattern. We suggest selecting only black and/or white when in 560 Mode.

### Program Sample:

```
10 & HGR : & MODE(2) : POKE - 1
    6301,0 : HOME : VTAB 21
20 PRINT ". [0].. [1].. [2].. [3].. [
    4].. [5].. [6].. [7].. [8].. [9].
    . [10]. [11]. [12]. [13]. [14]. [1
    5]BLACK.RED.DBLUE.VIO.DGRN.G
    REY1.MBLU.LBLU.BRN.ORNG.GREY
    2.PINK.BGRN.YELL.AQUA.WHTE"
30 FOR C = 1 TO 15: & HCOLOR= C
    : FOR X = C * 8.75 TO C * 8.
    75 + 8.75: & H PLOT X,0 TO X
    ,159 TO C,0: NEXT : NEXT
40 GET A$: HOME : & TEXT : LIST
```

## & HGR

**Syntax:** & HGR

**Example:** & HGR

**Purpose:** Converts the display to *split-screen* double hi-res graphics with a four-line text window at the bottom of the screen.

**Remarks:** This command will clear the screen to black and select the 80-column mode (sorry 40 column users; double hi-res graphics only works with 80-column text-screen text).

The & TEXT command will return display to the text screen.

To view full-screen graphics, POKE-16302,0.  
To switch back to split-screen, POKE-16301,0.

To plot in double hi-res while you are viewing the 80-column text screen, you need to POKE-16297,0. Then & PLOT, & LOAD, & SAVE, & CLEAR, etc. To see the image, type an & MODE command.

## & HGR2

**Syntax:** & HGR2

**Purpose:** Converts the display to *full-screen* double hi-res graphics with *no* text window at the bottom of the screen.

**Example:** & HGR2

**Remarks:** This command will clear the screen to black and select the 80-column mode (sorry, 40 column users, double hi-res graphics only works with 80-column text-screen text).

The & TEXT command will return to 80-column text display.

To view split-screen graphics, POKE-16301,0.  
To switch back to full-screen, POKE-16302,0.

**Note:** This command is different from Apple-soft's HGR2 command— **& HGR2** selects full-screen double hi-res, whereas **HGR2** selects page 2 of normal hi-res. See the Utilities section (later in the manual) for information on accessing double hi-res page 2.

## & HPLOT

**Syntax:** & HPLOT  $x_1,y_1$  [TO  $x_2,y_2\dots$ [TO  $x_n,y_n$ ]]  
& HPLOT TO  $x_2,y_2\dots$ [TO  $x_n,y_n$ ]

**Examples:** & HPLOT 75,20  
& HPLOT 48,115 TO 79,84 TO 110,115  
& HPLOT TO 125,10  
& HPLOT TO 0,0 TO 30,0 TO 30,30 TO 0,30

**Purpose:** Plots a point or draws a line on the double hi-res graphics screen.

**Remarks:**  $X_1, y_1, x_2, y_2, x_n$  and  $y_n$  must be numeric expressions within the screen limits.

The first syntax (above) can be used to plot either a single point or a series of lines.

The second syntax can be used to plot a series of lines starting from the last point plotted.

The color of the dot or line is determined by the last & HCOLOR command.

### Program Example:

```
10 & HGR : & MODE(2)
20 & HCOLOR= 15: & HPLOT 125,5
   6: REM WHITE DOT
30 & HCOLOR= 2: & HPLOT 0,50 TO
   139,50: REM BLUE LINE
40 & HPLOT TO 0,191: REM CONTI
   NUE LINE
50 & HCOLOR= 1: & HPLOT 0,0 TO
   139,0 TO 139,191 TO 0,191 TO
   0,0: REM RED BORDER
60 & MODE(1): & HCOLOR= 15: & HPLOT
   0,0 TO 559,191
100 GET A$: HOME : & TEXT : LIST
```



## & HSCRN

**Syntax:** & HSCRN (*x,y,variable*)

**Examples:** & HSCRN(50,75,A): PRINT A  
& HSCRN(100,0,B): PRINT B

**Purpose:** Returns the color value of the pixel at the specified (x,y) location.

**Remarks:** X and y must be numeric expressions within the screen limits.

*Variable* must be a real or integer variable. After executing the & HSCRN command, this variable will contain the color value of the pixel.

In 560 Mode, the value returned will always be zero or 15 (black or white).

In 140 Mode, the value returned will be 0-15, representing one of the 16 double hi-res colors (see & HCOLOR).

"& HSCRN" will list as "& H SCRN".

### Program Example:

```
10 & HGR : & MODE(2) : POKE - 1
    6301,0: HOME
15 X = 130:Y = 100: GOSUB 99
30 FOR I = 0 TO 15: & HCOLOR= I
    : & HPLOT 7 * I,0 TO 7 * I,
    157: NEXT
40 FOR I = 0 TO 15: VTAB 21: POKE
    36,I * 4: PRINT I: NEXT
45 PRINT : INPUT "HIT WHICH LINE
    ? (1-15):";L: GOSUB 99
60 & H SCRN( X,Y,V) : REM NOW V=C
    OLOR OF PIXEL X,Y
70 GOSUB 99: IF V < > L THEN GOSUB
    99:X = X - 1: GOTO 60
90 VTAB 1: PRINT CHR$( 7) : GET
    A$: HOME : & TEXT : LIST : END

99 & X PLOT X,Y TO X + 4,Y: RETURN
    : REM BULLET
```

## & LOAD

**Syntax:** & LOAD *main mem file,aux mem file*

**Examples:** & LOAD "PATTERNS","PATTERNS.AUX"  
& LOAD "PIC,D2","MYPIC.AUX,D2"  
(Notice quote marks around file names)

**Purpose:** Loads a double hi-res picture.

**Remarks:** Double hi-res pictures are loaded as two separate files, one for main memory and another for auxiliary memory.

*Main mem file* must be a string expression that specifies the file to load into main memory. This can be any legal file name and can include drive/slot parameters or Pro-DOS prefix.

*Aux mem file* must be a string expression that specifies the file to load into auxiliary memory. This can be any legal file name and can include drive/slot parameters or Pro-DOS prefix.

**Speed Tip:** See *ProntoDOS* on page 4.

### Program Example:

```
10 & HGR
30 HOME : VTAB 23: PRINT "For an
   example, try PATTERNS."
40 VTAB 21: INPUT "NAME OF PICTU
   RE TO BE LOADED: ";A$
50 B$ = A$ + ".AUX"
60 & HGR2
70 & LOAD A$,B$
80 GET A$: HOME : & TEXT : LIST
```

## & MODE

**Syntax:** & MODE(*mode number*)

**Examples:** & MODE(4)  
& MODE(1)

**Purpose:** Selects a double hi-res graphics mode.

**Remarks:** *Mode number* must be a numeric expression (1-4):

1: 560 Mode

2: 140 Mode

3: 560 Mixed Mode

4: 140 Mixed Mode

(See page 7.)

& MODE can be used instead of & HGR or & HGR2 to display the double hi-res screen *without* clearing the screen to black.

### Program Example:

```
10 & HGR2 : & HCOLOR= 15
20 & MODE(1) : & HPLOT 0,0 TO 13
    9,159
30 & MODE(2) : & HPLOT 0,0 TO 13
    9,159
40 POKE - 16301,0 : HOME : VTAB
    21 : HTAB 13 : PRINT "MODE(1) "
    ; SPC( 50) ; "MODE(2) "
50 GET A$ : HOME : & TEXT : LIST
```

## **& NORMAL**

**Syntax:** & NORMAL

**Example:** & NORMAL

**Purpose:** Turns off hi-res text, canceling the effect of & PRINT and & XPRINT.

**Remarks:** & PRINT changes the output hooks to allow double hi-res text. & NORMAL resets the hooks to their normal values to allow text-screen text and DOS commands.

This command will have no effect if the output hooks are already set to their normal values.

Use & NORMAL to quit printing in hi-res and start printing on the text screen.

# & PRINT

**Syntax:** & PRINT

**Example:** & PRINT  
SEE THE PROGRAM EXAMPLE ON THE FOLLOWING PAGE.

**Purpose:** Allows text output on the double hi-res screen using subsequent PRINT statements.

**Remarks:** Once & PRINT is executed, it will cause PRINT statements to print on the double hi-res screen.

**Important:** You must use & NORMAL before printing on the text screen and/or before using a DOS command (Catalog, Load, Save, etc.).

The only control-characters that can be "printed" are CHR\$(7) bell, CHR\$(8) backspace, CHR\$(10) linefeed, CHR\$(11) up-arrow, CHR\$(13) return and CHR\$(21) right-arrow. All other control-characters will be ignored.

**Text Size:** In 560 Mode [&MODE(1) or &MODE(3)], you can display narrow characters (80 columns x 24 lines). In 140 Mode [&MODE(2) or &MODE(4)], you can display wide characters (20 columns x 24 lines).

**Htab/Vtab:** HTAB and VTAB won't work. Use the & GOTO command (see program sample, next page).

**Color:** The double hi-res text color (foreground color) is changed with & HCOLOR. The background color is changed by & BCOLOR.

**Fonts:** Before using & PRINT you must load a hi-res font (or "character set") into memory (see program example, next page). Several fonts are included on the Beagle Graphics disk (files that end with ".FONT"). Apple's *DOS Toolkit* fonts are compatible with Beagle Graphics. *Apple Mechanic* fonts are not.

You must load fonts in a safe place (for example, above or below the hi-res screens,

## & PRINT (continued)

above himem, etc.). Once the font is loaded, you must tell DHGR where you put it. This is done by Poking the starting address of the font in to the 3rd and 4th bytes of DHGR using a lo-byte/hi-byte format.

### Program Example:

```
20 F = 16384: PRINT CHR$ (4); "BL
   OAD ASCII.FONT,A";F
30 L = PEEK (974) + PEEK (975) *
   256: REM DHGR LOCATION
40 POKE L + 3,0: POKE L + 4,64
100 & HGR2 : & PRINT : & HCOLOR=
   15: & B COLOR= 0
110 & MODE(1): & GOTO 0,0: PRINT
   "SMALL TYPE"
120 & MODE(2): PRINT "BIG TYPE"
125 PRINT : & HCOLOR= 1: PRINT
   "Color Type in": PRINT
127 & HCOLOR= 2: PRINT "DIFFERE
   NT STYLES!": & NORMAL
130 PRINT CHR$ (4); "BLOAD COLOS
   SAL.FONT": & PRINT
140 & HCOLOR= 12: PRINT "DIFFER
   ENT STYLES!": PRINT : PRINT
   : PRINT "PLUS..."
150 & GOTO 0,100:H = INT ( RND
   (1) * 16):B = INT ( RND (1)
   * 16): IF B < > H THEN &
   HCOLOR= H: & B COLOR= B: PRINT
   SPC( 15): PRINT : PRINT " M
   IXED COLORS! ": PRINT SPC(
   15)
155 IF PEEK ( - 16384) < 128 OR
   B = H THEN 150
160 HOME : & TEXT : & NORMAL :
   LIST
```

## & ROT

**Syntax:** & ROT= *rotation value*

**Examples:** & ROT=16  
& ROT=0

**Purpose:** Sets the rotation for double hi-res shapes to be drawn with & DRAW or & XDRAW.

**Remarks:** *Rotation value* must be a numeric expression 0-255. This number specifies the rotation in units of  $5.625^\circ$  (1/64th of a circle).

& ROT=0 causes the shape to be drawn upright, just as it was defined; & ROT=16 rotates the shape  $90^\circ$  clockwise; & ROT=32 rotates it  $180^\circ$ , and so on. To determine the rotation value, divide the number of degrees that you want to rotate the shape by 5.625. To rotate a shape  $270^\circ$  you would use 48 as the rotation value ( $270/5.625 = 48$ ).

### Program Example:

```
10 LOC = 24576: REM SHAPE TABLE LOCATION
15 POKE 232,0: POKE 233,96: REM POKE LOC INTO 232-233
20 FOR I = LOC TO LOC + 10: READ A: POKE I,A: NEXT
30 DATA 1,0,4,0,37,53,53,55,39,39,0
100 & SCALE= 9: & ROT= 0: & HGR : & MODE(2): & XDRAW 1 AT 50,100
105 FOR R = 0 TO 64: & ROT= R
110 & XDRAW 1 AT 50,100: & XDRAW 1 AT 50,100: NEXT
150 GET A$: HOME : & TEXT : LIST
```

## & SAVE

**Syntax:** & SAVE *main mem file,aux mem file*

**Examples:** & SAVE "PIC","PIC.AUX"

& SAVE A\$,B\$

& SAVE "FILE.1,S6,D2","FILE.2,S6,D2"

(Notice quote marks around file names.)

& SAVE A\$+DRIVE\$,B\$+DRIVE\$

**Purpose:** Saves a double hi-res picture to disk.

**Remarks:** Double hi-res pictures are saved as two separate files, one from main memory, and another from auxiliary memory.

*Main mem file* must be a string expression that specifies the file to be saved from main memory. This can be any legal file name and may include drive and slot parameters or ProDOS prefix.

*Aux mem file* must be a string expression that specifies the file to be saved from auxiliary memory. This can be any legal file name and may include drive and slot parameters or ProDOS prefix.

**Important:** Use the same name for the auxiliary memory file and the main memory file, except add ".AUX" to the auxiliary memory file name.

**The Picture-Save Shuffle:** A double hi-res image will "scramble" itself halfway through a save, then repair itself when finished. Don't panic; this is normal.

## WHAT'S AUXILIARY MEMORY?

We were hoping you wouldn't ask. Just think of "main memory" as your Apple's first 64K and auxiliary memory as its second 64K. Each half of a double hi-res picture must be displayed from different parts of memory.



## & SCALE

**Syntax:** & SCALE = *scale factor*

**Examples:** & SCALE=1  
& SCALE=5

**Purpose:** Sets the scale (size) for double hi-res shapes to be drawn with & DRAW or & XDRAW.

**Remarks:** *Scale factor* must be a numeric expression 0-255.

& SCALE=1 will cause a point-for-point reproduction of the shape. & SCALE=2 doubles the shape's size, and so on. & SCALE=0 is equivalent to & SCALE=255.

### Program Example:

```
10 LOC = 24576: REM SHAPE TABLE LOCATION
15 POKE 232,0: POKE 233,96: REM POKE LOC INTO 232-233
20 FOR I = LOC TO LOC + 10: READ A: POKE I,A: NEXT
30 DATA 1,0,4,0,37,53,53,55,39,39,0
100 & ROT= 0: & HGR : & MODE(2): & HCOLOR= 15
105 FOR S = 1 TO 36: & SCALE= S
110 & DRAW 1 AT 20,100: NEXT
150 GET A$: HOME : & TEXT : LIST
```

## & TEXT

**Syntax:** & TEXT

**Example:** & TEXT

**Purpose:** Returns display to the text screen.

**Remarks:** This command is used to switch from the graphics display back to the text display. This is the only proper way to exit DHGR.

Note: Regular hi-res commands will not work properly if you don't exit DHGR by using & TEXT.

### Program Example:

```
10 & HGR2 : & MODE(1) : & HCOLOR=
    15: POKE - 16301,0
15 HOME : LIST
20 & H PLOT 0,0 TO 559,0 TO 559,
    159 TO 0,159 TO 0,0: & CIRCL
    E(279,79) AT 279,79
30 VTAB 21: PRINT : PRINT "THIS
    IS THE GRAPHICS SCREEN.";
40 PRINT : PRINT "PRESS ANY KEY
    (OR <RETURN> TO QUIT): ";; GET
    A$: & TEXT
50 VTAB 21: PRINT : PRINT "THIS
    IS THE TEXT SCREEN."; SPC( 4
    )
55 IF A$ = CHR$(13) THEN 80
60 PRINT : PRINT "PRESS ANY KEY
    (OR <RETURN> TO QUIT): ";; GET
    A$: & MODE(1)
70 POKE - 16301,0: IF A$ < > CHR$(
    13) THEN 30
80 HOME : & TEXT : LIST
```

## & XBOX

**Syntax:** & XBOX (*x length* [,*y length*]) [AT *x,y*]

**Examples:** & XBOX (50,25) AT 80,80  
& XBOX (50) AT 70,80  
& XBOX (100)

**Purpose:** Draws a square or rectangle on the double hi-res screen. This command works just like the & BOX command, except that each point of the box is plotted using the complement (opposite) of the color already displayed at that point.

**Remarks:** This command is most commonly used to erase a previously drawn box.  
See & BOX for rules regarding the use of & XBOX.

### Program Example:

```
100 & HGR :X = 40:Y = 20
120 & XBOX(X,Y) AT 50,50: REM DR
    AW IT
130 HOME : VTAB 22: PRINT "<G> T
    O GROW / <S> TO SHRINK / <Q>
    TO QUIT: ";: GET A$: PRINT
    A$
140 & XBOX(X,Y) AT 50,50: REM E
    RASE IT
145 IF A$ = "G" THEN X = X + 10
150 IF A$ = "S" THEN X = X - 10
155 IF A$ = "Q" THEN 200
160 Y = X / 2: GOTO 120
200 HOME : & TEXT : LIST
```

## & XCIRCLE

**Syntax:** & XCIRCLE (*x radius* [,*y radius*]) [AT *x,y*]

**Examples:** & XCIRCLE (20,10) AT 100,100  
& XCIRCLE (40) AT 70,96  
& XCIRCLE (100)

**Purpose:** Draws a circle or ellipse on the double hi-res screen. This command works just like & CIRCLE, except that each point of the circle is plotted using the complement (opposite) of the color already displayed at that point.

**Remarks:** & XCIRCLE is most commonly used to erase a previously drawn circle.

See & CIRCLE for rules regarding the use of & XCIRCLE.

### Program Example:

```
100 & HGR :X = 280:Y = 80
120 & XCIRCLE(80,40) AT X,Y: REM
    DRAW IT
130 HOME : VTAB 22: PRINT "ARROW
    KEY TO MOVE, OR <Q> TO QUIT
    : ";: INVERSE : PRINT SPC(
    1): NORMAL
135 K = PEEK ( - 16384): IF K <
    128 THEN 135
137 POKE - 16368,0:K = K - 128
140 & XCIRCLE(80,40) AT X,Y: REM
    ERASE IT
145 IF K = 8 THEN X = X - 10
150 IF K = 21 THEN X = X + 10
160 IF K = 10 THEN Y = Y + 10
170 IF K = 11 THEN Y = Y - 10
180 IF K < > 81 THEN 120
200 HOME : & TEXT : LIST
```

## & XDRAW

**Syntax:** & XDRAW *shape number* [AT *x,y*]

**Examples:** & XDRAW 4 AT 50,100  
& XDRAW 42

(assumes a shape table is loaded and pointers are set)

**Purpose:** Draws a shape at the specified location on the double hi-res screen from the shape table currently in memory. This command works the same as & DRAW, except that each point in the shape is plotted using the complement (opposite) of the color currently displayed at that point.

**Remarks:** This command is most commonly used to erase a previously drawn shape.

See & DRAW for rules regarding the use of & XDRAW.

### Program Example:

```
10 LOC = 24576: REM SHAPE TABLE L
   OCATION
15 POKE 232,0: POKE 233,96: REM
   POKE LOC INTO 232-233
20 FOR I = LOC TO LOC + 10: READ
   A: POKE I,A: NEXT
30 DATA 1,0,4,0,37,53,53,55,39,3
   9,0
100 & SCALE= 9: & ROT= 0: & HGR
   : & MODE(2)
110 FOR X = 0 TO 100 STEP 5
120 & XDRAW 1 AT X,100: & XDRAW
   1 AT X,100
130 NEXT : & XDRAW 1 AT X,100
150 GET A$: HOME : & TEXT : LIST
```

## & XPLOT

**Syntax:** & XPLOT *x1,y1* [TO *x2,y2...* [TO *xn,yn*]]  
& XPLOT TO *x2,y2...*[TO *xn,yn*]

**Examples:** & XPLOT 75,20  
& XPLOT 48.115 TO 79,84 TO 110,115  
& XPLOT TO 125,10

**Purpose:** Plots a point or draws a line on the double hi-res screen. This command works just like the & HPLOT command, except that each point or line is plotted using the complement (opposite) of the color currently display at that point or line.

**Remarks:** & XPLOT is most commonly used to erase a previously drawn point or line.

See & HPLOT for rules regarding the use of & XPLOT.

"& XPLOT" will list as "& X PLOT".

"Complement", in this case, doesn't mean *color-wheel* complement. Run the example program below and you'll see.

### Program Example:

```
10 & HGR : & MODE(2) : POKE - 1
    6301,0 : HOME : VTAB 21
20 FOR C = 1 TO 15 : & HCOLOR= C
    : FOR X = C * 8 TO C * 8 + 4
    : & HPLOT X,50 TO X,99 : NEXT
    : NEXT
30 FOR Y = 70 TO 80 : & X PLOT 0,
    Y TO 139,Y : NEXT
40 GET A$: HOME : & TEXT : LIST
```

## & XPRINT

**Syntax:** & XPRINT

**Example:** & XPRINT

**Purpose:** Allows text output anywhere on the double hi-res screen. This command works just like & PRINT, except that the color of each character will be the complement (opposite) of the existing color.

**Remarks:** & XPRINT is commonly used to erase a character already printed or to print on top of another color.

See & PRINT for rules regarding the use of & XPRINT.

"& XPRINT" will list as "& X PRINT".

### Program Example:

```
20 F = 16384: PRINT CHR$(4);"BL
   OAD ASCII.FONT,A";F
30 L = PEEK(974) + PEEK(975) *
   256
40 POKE L + 3,0: POKE L + 4,64
100 & HGR2 : & MODE(2)
130 FOR N = 1 TO 6: READ A,B
140 DATA 1,20,15,30,2,40,1,70,15
   ,80,12,90
145 & HCOLOR= A: FOR I = B TO B
   + 10: & HPLOT I,50 TO I,12
   0: NEXT : NEXT
150 FOR I = 1 TO 13: GOSUB 200: NEXT
160 GET A$: HOME : & TEXT : & NORMAL
   : LIST : END
200 & X PRINT : & GOTO 15,100: PRINT
   "FRANCE MEXICO": RETURN
```





# DGR (alias Double lo-res Graphics)



DGR is a machine language program that adds eight double Low Resolution commands to Applesoft.

## LOADING DGR

To install DGR in memory, type "**BRUN DGR**" directly from the keyboard, or put this DOS command in your Applesoft program:

**10 PRINT CHR\$(4);"BRUN DGR"**

After loading DGR, the following commands can be used in your Applesoft programs:

<i>New command</i>	<i>Normal lo-res equivalent</i>
<b>&amp; GR</b>	<b>GR</b>
<b>&amp; GR2</b>	(none)
<b>&amp; PLOT</b>	<b>PLOT</b>
<b>&amp; HLIN</b>	<b>HLIN</b>
<b>&amp; VLIN</b>	<b>VLIN</b>
<b>&amp; COLOR</b>	<b>COLOR</b>
<b>&amp; SCRN</b>	<b>SCRN</b>
<b>&amp; TEXT</b>	<b>TEXT</b>

Technical information may be found on previous pages:

Trouble Shooting . . . . .	33
Loading procedures and problems . . . . .	35
Combining DGR with other & programs . . . . .	35
Command Description Format . . . . .	36

## DOUBLE LO-RES SCREEN LAYOUT

A double-res screen layout chart appears on the back of the Help card that came with your Beagle Graphics disk. When using lo-res commands, X (horizontal) must be a numeric expression 0-79. Y (vertical) must be a numeric expression 0-47 (full-screen graphics), or 0-39 (split-screen with four lines of text at the bottom).

## & COLOR

**Syntax:** & COLOR=*color value*

**Examples:** & COLOR=11  
& COLOR=15

**Purpose:** Sets the display color for plotting double lo-res graphics.

**Remarks:** *Color value* is a numeric expression 0-15:

0 Black	8 Brown
1 Magenta (Red)	9 Orange
2 Dark Blue	10 Grey 2
3 Violet	11 Pink
4 Dark Green	12 Green
5 Grey 1	13 Yellow
6 Medium Blue	14 Aqua
7 Light Blue	15 White

(See the colors on the Help card that came with your disk.)

*Color value* will be used for plotting all blocks (double lo-res *points*) and lines.

*Color values* greater than 15 and smaller than 255 are treated as "mod 16" values; 16=0, 17=1, etc.

### Program Example:

```
10 & GR : HOME
20 FOR C = 0 TO 15: & COLOR= C
30 FOR I = 0 TO 4: & VLIN 0,39 AT
   C * 5 + I: NEXT : NEXT
40 FOR C = 0 TO 15: VTAB 21: POKE
   36,5 * C + 2: PRINT C: NEXT
50 GET A$: & TEXT : HOME : LIST
```

**Important:** If no & COLOR is specified after an & GR or & GR2 command, the default color zero (black) will be used.

**Also:** If you try drawing in double lo-res while viewing the text screen, a *character* will be displayed instead of a block. Always use & GR or & GR2 before executing & PLOT, & HLIN and & VLIN commands.

## **& GR**

**Syntax:** & GR

**Example:** & GR

**Purpose:** Converts the screen display to split-screen double lo-res graphics (80 x 40 pixels) with a four-line text window at the bottom of the screen (Vtab 21-24).

**Remarks:** & GR clears the screen to black and sets & COLOR to zero (black).

The & TEXT command will return to text screen display.

To select full-screen display (80 x 48 pixels) without clearing the screen, POKE -16302,0 (& GR2 does the same, but erases the screen). POKE-16301,0 will switch back to split-screen without erasing.

## & GR2

**Syntax:** & GR2

**Example:** & GR2

**Purpose:** Converts the screen display to full-screen double lo-res graphics (80 x 48 pixels) with no text window at the bottom of the screen.

**Remarks:** & GR2 clears the screen to black and sets & COLOR to zero (black).

The & TEXT command will return to text screen display.

To select a split-screen display (80 x 40 pixels) without clearing the screen, type POKE-16301,0 (& GR does the same, but erases the screen). POKE-16302,0 will switch back to full-screen without erasing.

"& GR2" lists as "& GR 2".

## & HLIN

**Syntax:** & HLIN *x1,x2* AT *y*

**Examples:** HLIN 0,79 AT 0  
HLIN 20,25 AT 25

**Purpose:** Draws a horizontal double lo-res line.

**Remarks:** *X1* and *x2* specify the start- and end-column of the line to be drawn. They must be numeric expressions 0-79.

"AT *y*" specifies the row at which the line is to be drawn and must be a numeric expression 0-47 (or 0-39 if the text window is displayed).

The color of the line is determined by the last & COLOR statement.

### Program Example:

```
10 & GR 2:C = INT ( RND (1) *
    15)
20 FOR Y = 0 TO 47: & COLOR= C:
    & HLIN 0,79 AT Y
30 X = X + 1: IF Y > 38 THEN X =
    2
40 & COLOR= INT ( RND (1) * 15
    ): & HLIN 40 - X,40 + X AT
    Y
45 NEXT
50 GET A$: & TEXT : HOME : LIST
```

## & PLOT

**Syntax:** & PLOT x,y

**Examples:** & PLOT 0,0  
& PLOT 79,47

**Purpose:** Plots a single block (double lo-res *point*) at the specified x,y position on the double lo-res screen.

**Remarks:** X must be a numeric expression 0-79. This specifies the horizontal position of the block. Y must be a numeric expression 0-47 (or 0-39 if the text window is displayed at the bottom of the screen).

The color of the block is determined by the last & COLOR statement. If no color has been specified, the default color zero (black) will be used.

### Program Example:

```
10 & GR : HOME : & COLOR= 15
20 FOR Q = 0 TO 39: & PLOT Q,Q:
    & PLOT Q,39 - Q: & PLOT 4
    0 + Q,Q: & PLOT 40 + Q,39 -
    Q: NEXT
30 VTAB 22: GET A$: & TEXT : HOME
    : LIST
```

## & SCRN

**Syntax:** & SCRN (*x,y,variable*)

**Examples:** & SCRN(5,10,A)  
& SCRN(79,0,HUE)

**Purpose:** Returns the color value of the block at the designated position on the double lo-res screen.

**Remarks** X specifies the horizontal position of the block and must be a numeric expression 0-79. Y specifies the vertical position and must be a numeric expression 0-47 (or 0-39 if the text window is displayed at the bottom of the screen).

*Variable* is a real or integer variable. After executing the &SCRN command, this variable will contain the color value of the block at x,y. See & COLOR for the list of color values.

### Program Example:

```
10 & GR : HOME
15 X = 75:Y = 30: GOSUB 99
30 FOR I = 0 TO 15: & COLOR= I:
    & VLIN 0,39 AT I * 4: NEXT
40 FOR I = 1 TO 15: VTAB 21: POKE
    36,I * 4: PRINT I: NEXT
50 PRINT : INPUT "HIT WHICH LINE
    ? (1-15): ";L
60 & SCRN( X,Y,V): REM NOW V=CO
    LOR OF PIXEL X,Y
70 GOSUB 99: IF V < > L THEN GOSUB
    99:X = X - 1: GOTO 60
90 PRINT CHR$( 7);: GET A$: & TEXT
    : HOME : LIST : END)
99 & HLIN X,X + 4 AT Y: RETURN
    : REM BULLET
```

## & TEXT

**Syntax:** & TEXT

**Example:** & TEXT

**Purpose:** Returns to the text display.

**Remarks:** This command is used to switch from graphics display back to the text display. This is the *only* proper way to exit DGR.

Regular lo-res commands will not function properly if you don't exit DGR by using & TEXT.

### Program Example:

```
10 & GR : & COLOR= 15
20 & HLIN 0,79 AT 0: & HLIN 0,
   79 AT 39: & VLIN 0,39 AT 0:
   & VLIN 0,39 AT 79
30 HOME : VTAB 22: PRINT "THIS I
   S THE GRAPHICS SCREEN.": GET
   A$
40 HOME : & TEXT : VTAB 22: PRINT
   "THIS IS THE TEXT SCREEN.": GET
   A$
50 RUN
```



## & VLIN

**Syntax:** & VLIN y1,y2 AT x

**Examples:** & VLIN 5,15 AT 45  
& VLIN 0,39 AT 79

**Purpose:** Draws a vertical line on the double lo-res screen.

**Remarks:** Y1 and y2 specify the start- and end-row of the line to be drawn, and must be a numeric expression 0-47 (or 0-39 if the text window is displayed).

"AT x" specifies the column at which the line is to be drawn, and must be a numeric expression 0-79.

The color of the line is determined by the last & COLOR statement.

### Program Example:

```
10 & GR : HOME : & COLOR= 15
20 VTAB 21: FOR I = 0 TO 25: PRINT
   CHR$ (65 + I); SPC( 2);: NEXT
30 FOR I = 0 TO 25: & VLIN 39,3
   9 - INT ( RND (1) * 39) AT
   3 * I: NEXT
40 PRINT : PRINT : GET A$: & TEXT
   : HOME : LIST
```



# FONT.EDITOR



The Beagle Graphics FONT.EDITOR program lets you redraw individual double hi-res characters or entire character sets. You may create different versions of a font with special symbols, or redraw an entire font from scratch. Altered fonts may be saved on disk and be used from Double Plot's Text Mode, or from within your own programs using the & PRINT command followed by PRINT statements.

To use the Font Editor, select option F from the boot menu that appears when you boot the Beagle Graphics disk. Or, type:

## **RUN FONT.EDITOR**

You will first be asked to type the name of the character set to be edited. Type the font's name to load it. OR press [RETURN] to see a catalog of the disk. Beagle Graphics' character sets have names that end in ".FONT". You will probably want to save your edited version back on the disk using a different name. We recommend keeping the ".FONT" suffix, however.

Apple's *DOS Tool Kit* fonts are compatible with this program, and so are many others. Beagle Bros' *Apple Mechanic* shape table fonts are not. Sorry.

After selecting a character set to edit, you will see the Font Editor Screen. On the left is the character set that is in memory, with its name below. Above is an indication of the current mode, SELECT or EDIT. The center of the screen contains a rectangular plotting grid used for drawing characters. The dotted lines within the grid represent the 7 x 8 matrix in which all characters are formed.

## **SELECT MODE**

This mode allows you to load and save character sets and select the individual characters that you want to edit or replace. Use the four **ARROW KEYS** to move the cursor to the desired character. As you move, the ASCII value of the character under the cursor and the actual key that will type that character will be displayed at the right of the character set (any key may type any symbol). Press **[RETURN]** when the cursor is at the character you want to edit. This will place you in EDIT MODE.

**Type S** to save the current character set to disk or to load a different font. An option is provided to save the character set using a different name than the original.

**Control-Q** quits the program.

## **EDIT MODE**

This mode allows you to alter or re-draw the character chosen from the Select Mode (above). The character will be displayed in enlarged format with a plotting cursor (a cross). Modify the character by using the four **ARROW KEYS** to move the cursor and the **SPACE BAR** to plot or unplot the pixel under the cursor. You will see an actual-size version of the symbol you are drawing in the character set table.

Press **[RETURN]** when you are finished editing the character. This will return you to the Select Mode, allowing you to select another character, save the character set, or quit.

# SLIDE.SHOW



This little program allows you to create "slide show" presentations of your double hi-res pictures. Start by selecting option S from the boot menu, or by typing:

## **RUN SLIDE.SHOW**

The program is initially set up to display all of the sample pictures on the Beagle Graphics disk. Watch it work and then LIST the program. Each picture will be displayed a pre-programmed length of time before moving to the next picture.

Skip to the next picture (overriding the delay) by pressing the Right-Apple (*Closed-Apple*) key (or joystick Button #1). Left-Apple (that's *Open-Apple* key or Button #0) backs up to the previous picture.

Press [ESC] to quit a slide show.

Here's what you do to make your own slide show:

**1. Transfer all of the pictures** to be displayed onto the same disk with these programs:

SLIDE.SHOW

DOUBLE.SCRUNCH

DHGR

Use DOS 3.3's FID program or ProDOS's FILER to make the transfer (see your DOS instructions). Please use a relatively empty disk.

**2. LOAD SLIDE.SHOW** and designate which pictures to display and the length of time each one should be on the screen. This is done by adding Data statements to the program starting at line 150. Each Data statement must consist of a picture name and its display time, in seconds. For example:

```
150 DATA HOUSE,5
```

```
160 DATA DOG,14
```

```
170 DATA LUNCH,9
```

“AUX” files should not be included in the Data statements, but they must, of course, be on the disk. The pictures will be displayed in the order they appear in the Data statements.

The only limit to the number of pictures is the number that you can fit on a disk (along with the other required programs from step 1). You may store more pictures on a disk by using the DOUBLE.SCRUNCH program to “compress” pictures. File names for compressed pictures *must* end with “.PAC” (for example, “HOUSE.PAC”) to work with the SLIDE.SHOW program.

**3. SAVE your modified program** on disk and you’re ready to roll—lights, popcorn, action!

Feel free to add enhancements to SLIDE.SHOW, like displaying double lo-res pictures or text screens, controlling the slide show via mouse or joystick, etc.



## DOUBLE-RES UTILITIES



Several graphics utilities are included on the Beagle Graphics disk, written in machine-language and accessible directly from the keyboard and (most) from within your Applesoft programs using CALL statements.

Here is a rundown of the utilities and what they do. The "A" and "L" figures are the start and length of each program. If you don't understand this, you don't need to.

### DOUBLE HI-RES UTILITIES

**CHANGE.COLORS:** Changes any double hi-res color on the screen to any other. (A\$6000,L\$146)

**CONVERT.HIRES.1:** Converts normal hi-res pictures to half-width double hi-res. (A\$300,L\$76)

**CONVERT.HIRES.2:** Converts normal hi-res pictures to full-width double hi-res. (A\$300,L\$69)

**CUT.AND.PASTE:** Lets you save and move sections of double hi-res pictures. (A\$6000,L\$147)

**DOUBLE.SCRUNCH:** Packs (and unpacks) double hi-res picture data to save disk space. (A\$7000,L\$1D5)

**HGR.TO.DHGR:** Converts normal hi-res programs to double hi-res by adding an "&" to each hi-res command. (A\$300,L\$AA)

**PAGE.2:** Simulates double hi-res page 2. (A\$300,L\$CA)

### DOUBLE LO-RES UTILITIES

**CONVERT.LORES.1:** Converts normal lo-res pictures to half-width double lo-res. (A\$300,L\$8E)

**CONVERT.LORES.2:** Converts normal lo-res pictures to full-width double lo-res. (A\$300,L\$7F)

**GR.TO.DGR:** Converts normal lo-res programs to double lo-res by adding an "&" to each lo-res command. (A\$300,L\$A7)

**LORES.LOAD.SAVE:** Lets you load and save double lo-res images and 80-column screens on disk. (A\$300,L\$AA)



Instructions for the Beagle Graphics utilities appear on the following pages. But first, a few rules...

## **UTILITY NOTES**

- You supply all of the lower case items. For example, if the instructions say type "BLOAD picture", you type "BLOAD" followed by your picture's file name.
- You may skip the "Poke-16301,0" split-screen and "Vtab 21" cursor-positioning commands if you are using a utility from within a program.
- If you don't know much about "memory addresses" and the like, substitute the number 16384 for "tempaddr". Technical information coming up...

## **WHAT'S A TEMPADDR?**

Many of the utility instructions refer to a temporary address, or "*tempaddr*". This is a number, supplied by you, that designates a memory location used to temporarily hold data. You must pick a location that won't interfere with your program and its variables, the double hi-res screen, etc.

**IF YOU ARE WRITING STRAIGHTFORWARD NORMAL-SIZE PROGRAMS, SUBSTITUTE 16384 FOR "tempaddr".** This specifies hi-res page 2, address 16384 (hex \$4000).

If your program is above hi-res page 1, then address 2048 (\$800 hex) is usually a safe value for tempaddr.

DON'T use 768 (\$300) or 24576 (\$6000). The utilities use these locations.

## CHANGE.COLORS

This utility changes any color in a double hi-res picture to any other color. Use the following procedure directly from the keyboard or from within a program:

1. Load DHGR (if it isn't already loaded):  
**BRUN DHGR**
2. Load the utility:  
**BLOAD CHANGE.COLORS**
3. Select 140 mode:  
**& HGR: & MODE(2)**
4. Select split-screen (graphics & text) and move the cursor into view:  
**POKE -16301,0: VTAB 21**
5. Load your picture if necessary:  
**& LOAD "picture","picture.AUX"**
6. Poke the value of the color to be changed. (See your Help card for a list of color values). Substitute the color's number for "oldcolor":  
**POKE 212,oldcolor**
7. Set & HCOLOR to the new color value:  
**& HCOLOR=newcolor**
8. Call the color conversion routine:  
**CALL 24576**

# CONVERT.HIRES.1

This utility converts normal hi-res pictures to double hi-res. It will duplicate the picture pixel-for-pixel on the *left half\** of the double hi-res screen. Use the following procedure directly from the keyboard or within a program:

1. Load DHGR (unless it's already loaded):  
**BRUN DHGR**
2. Select the double hi-res screen and move the cursor into view:  
**& HGR : VTAB 21**
3. Load the conversion utility:  
**BLOAD CONVERT.HIRES.1**
4. Load the hi-res picture to be converted to double hi-res. It *must* be loaded at address \$4000 (hi-res page 2). Substitute your picture name for "picture":  
**BLOAD picture,A\$4000**
5. Call the conversion routine:  
**CALL 768**
6. The converted picture will appear on the screen. If you like what you see, use the & SAVE command to save it to disk:  
**& SAVE "picture","picture.AUX"**

\* **Note:** You can move the picture from the left side of the screen using Double Plot or the CUT.AND.PASTE utility.

## CONVERT.HIRES.2

This utility converts normal hi-res pictures into double hi-res pictures by doubling each pixel. This method will preserve a picture's proportions and some of its colors. Use the following procedure directly from the keyboard or from within a program:

1. Load DHGR (unless it's already loaded):  
**BRUN DHGR**
2. Load the utility:  
**BLOAD CONVERT.HIRES.2**
3. Select the double hi-res screen and move the cursor into view:  
**& HGR : VTAB 21**
4. Load a normal hi-res picture. This picture *must* be loaded at address \$4000 (hi-res page 2). Substitute your picture's name for "picture":  
**BLOAD picture,A\$4000**
5. Call the conversion routine:  
**CALL 768**
6. The converted picture will appear on the screen. If you like what you see, use the & SAVE command to save it to disk:  
**& SAVE "picture","picture.AUX"**

## CUT.AND.PASTE

This utility is used to “cut” an area from the double hi-res screen and then “paste” it back at another location, *OR* to save only the cut area to disk to save disk space.

To cut an area from the screen, you must specify two *diagonal corners* of the area—(x1,y1) and (x2,y2).

**IMPORTANT:** For this utility only, the screen is divided into **80 horizontal blocks** (x-axis: 0-79) and 192 vertical rows (y-axis: 0-191). Each horizontal block is 7 pixels wide x 1 pixel high (7 pixels x 80 blocks = 560 pixels). For example, if you specified (0,0) and (5,50) as diagonal corners of an area, it would be a box 35 (5 times 7) pixels wide by 50 pixels high.

### CUT

To specify a picture area to be moved or saved, follow this procedure (typed directly or from within a program):

1. Load DHGR (unless it's already loaded):

**BRUN DHGR**

2. Load the utility:

**BLOAD CUT.AND.PASTE**

3. Select double hi-res and move the cursor into view:

**& HGR : VTAB 21**

4. Poke the temporary storage address. Substitute the number 16384 for tempaddr if you want (see page 80):

**POKE 0,tempaddr-INT(tempaddr/256)\*256**

**POKE 1,INT(tempaddr/256)**

5. Poke the starting block (x1), starting row (y1), ending block (x2), and ending row (y2):

**POKE 24576+6,x1: POKE 24576+7,y1:**

**POKE 24576+8,x2: POKE 24576+9,y2**

6. Call the routine to cut the area from the screen (and place it in the temporary storage area):

**CALL 24576**

(The cut area will not be erased; just stored in memory.)

7. Print the length of the area cut from the screen

**PRINT PEEK(0)+PEEK(1)\*256-tempaddr**

(continued)

8. If you want to save the cut area to disk, type this command using the length value from step 7:

**BSAVE blockname ,A tempaddr ,L length**

(For example, BSAVE BLOCK,A16384,L1234)

Skip this step if you just want to paste the area in a different location or picture.

## **PASTE**

To paste a cut area onto the screen, do this:

1. Load DHGR (unless it's already loaded):

**BRUN DHGR**

2. Load the utility (if it isn't already loaded):

**BLOAD CUT.AND.PASTE**

3. Select double hi-res and move the cursor into view:

**& HGR : VTAB 21**

4. Poke the address of the temporary storage area (this must be done *each time* you paste). Substitute the number 16384 for tempaddr if you want (see page 80):

**POKE 0,tempaddr-INT(tempaddr/256)\*256**

**POKE 1,INT(tempaddr/256)**

5. Load the area to be pasted into the temporary storage area. Skip this step if the area is already in temporary storage (if you just cut it):

**BLOAD blockname, A tempaddr**

6. Poke in the new (x,y) location where you want the area to be placed. REMEMBER: You are using 7-pixel wide *blocks* instead of individual pixels—X must be between 0 and 79. Skip this step if you want to “paste” the area at the same (x,y) location it was “cut” from:

**POKE tempaddr,x**

**POKE tempaddr+1,y**

7. Call the paste routine:

**CALL 24576+3** (or CALL 24579)

# DOUBLE.SCRUNCH

This utility helps postpone those nasty DISK FULL error messages by "compressing" the data used by double hi-res pictures. "Packed" pictures occupy less disk space, take up only one filename position in catalogs *and* they load faster. Here's the procedure:

## TO PACK A PICTURE:

1. Load DHGR (unless it's already loaded):  
**BRUN DHGR**
2. Load the utility:  
**BLOAD DOUBLE.SCRUNCH**
3. Select double hi-res and move the cursor into view:  
**& HGR: VTAB 21**
4. Poke in the lo-byte and hi-byte values of the temporary storage area address. Substitute the number 16384 for *tempaddr* if you want:  
**POKE 0,tempaddr-INT(tempaddr/256)\*256**  
**POKE 1,INT(tempaddr/256)**
5. Load your double hi-res picture:  
**& LOAD "picture","picture.AUX"**
6. Call the compress routine. (The compressed form of the picture will be placed in the temporary storage area):  
**CALL 28672**
7. Print the length of the compressed picture:  
**PRINT PEEK(0)+PEEK(1)\*256-tempaddr**
8. Save the compressed picture using the suffix ".PAC", *tempaddr* and the length value from step 7:  
**BSAVE picture.PAC, A tempaddr ,L length**  
(For example, BSAVE HOUSE.PAC,A16384,L1234)

**Note:** Compressed pictures are saved as a single file. Uncompressed double hi-res pictures are saved as two files.

(continued)

### To UNPACK a picture:

1. Load DHGR (unless it's already loaded):  
**BRUN DHGR**
2. Load the utility:  
**BLOAD DOUBLE.SCRUNCH**
3. Select double hi-res and move the cursor into view:  
**& HGR: VTAB 21**
4. Poke in the lo-byte and hi-byte values of the address for the temporary storage area. Substitute the number 16384 for tempaddr if you want (see page 80):  
**POKE 0,tempaddr-INT(tempaddr/256)\*256:**  
**POKE 1,INT(tempaddr/256)**
5. Load the compressed picture that you want to unpack:  
**BLOAD picture.PAC,A tempaddr**
6. Call the unpack routine:  
**CALL 28672+3** (or simply **CALL 28675**)

### DOUBLE.SCRUNCH EXAMPLES

These commands, typed directly from the keyboard, will pack the Beagle Graphics PATTERNS picture:

```
BRUN DHGR (only if not already loaded)  
BLOAD DOUBLE.SCRUNCH  
& HGR: VTAB 21: TEMPADDR=16384  
POKE 0,TE-INT(TE/256)*256  
POKE 1,INT(TE/256)  
& LOAD "PATTERNS","PATTERNS.AUX"  
CALL 28672  
PRINT PEEK(0)+PEEK(1)*256-TE  
Apple prints 10335  
BSAVE PATTERNS.PAC,A16384,L10335
```

This program will unpack the BEAGLE.PAC picture:

```
10 PRINT CHR$(4);"BRUN DHGR" (only if not loaded)  
20 PRINT CHR$(4);"BLOAD DOUBLE.SCRUNCH"  
30 TEMP=16384  
40 POKE 0,TE-INT(TE/256)*256: POKE 1,INT(TE/256)  
50 &HGR2: PRINT CHR$(4);"BLOAD BEAGLE.PAC"  
60 CALL 28675
```



# HGR.TO.DHGR

This utility converts regular hi-res programs to double hi-res by inserting an ampersand (&) before each hi-res command found in the program ("HPLOT" becomes "& HPLOT", etc.). Type the commands below directly from the keyboard (*not* in a program):

1. Load the HGR.TO.DHGR utility:

**BLOAD HGR.TO.DHGR**

2. Load the Applesoft program to be converted. Substitute your program name for "program".

**LOAD program**

3. Call the ampersand inserter routine:

**CALL 768**

Your program will be converted. Save the altered program using a new name.

## PROGRAM NOTES

- An ampersand will be placed in front of all hi-res commands unless doing so would create a line that is too long for Applesoft.
- Double hi-res's & HGR2 performs a different function than regular hi-res's HGR2. You will have to adjust your program to compensate.
- You will have to change color values yourself. For example, in hi-res, white is color #4 or #7. In double hi-res, white is color #15.

## PAGE.2

This utility is used to *simulate* normal hi-res page 2, which doesn't really exist in double hi-res graphics. You may do three things with this utility:

- Move a double hi-res picture out of sight to page 2, for temporary storage (erases previous page 2 image).
- Move a picture from page 2 into view on page 1 (erases previous page 1 image).
- Swap page 1 and page 2.

You *cannot* "page flip" with double hi-res pictures as you can with normal hi-res. If you don't know what page-flipping is... well, it's another subject.

The following procedure leaves out the steps of loading pictures, clearing the screen, etc.:

1. Load the PAGE.2 utility:

### **BLOAD PAGE.2**

2. Do one of the following:

**CALL 768** to move page 1 to page 2.

**CALL 768+3** to move page 2 to page 1.

**CALL 768+6** to swap pages 1 and 2.



# CONVERT.LORES.1

This utility converts an existing lo-res picture to double lo-res, duplicating the image pixel-for-pixel on the *left half\** of the double lo-res screen. Use the following procedure directly from the keyboard or from within a program:

1. Load DGR (unless it's already loaded):  
**BRUN DGR**
2. Select double lo-res graphics:  
**& GR**
3. Load the conversion utility:  
**BLOAD CONVERT.LORES.1**
4. Load the lo-res picture to be converted. Substitute the appropriate name for "picture":  
**BLOAD picture ,A\$4000**
5. Call the conversion routine:  
**CALL 768**

The picture is now converted. Save the new double lo-res picture using the LORES.LOAD.SAVE utility.

CONVERT.LORES.1 will compress the picture on the left side of the screen. See CONVERT.LORES.2 (next page) for another method.

\* This program will move a double lo-res picture horizontally across the screen; slowly but effectively:

```
10 D = 1: REM DIRECTION (0=LEFT, 1=RIGHT)
20 N = 40: REM NUMBER OF COLUMNS
   TO MOVE (MAX=40, MIN=1)
30 IF NOT D THEN 60
40 FOR Y = 0 TO 39: REM SPLIT=39, FULL=47
50 FOR X = 79 - N TO 0 STEP - 1
   : & SCRN( X,Y,Z): & COLOR=
   Z: & PLOT X + N,Y: & COLOR=
   0: & PLOT X,Y: NEXT X,Y: END
60 FOR Y = 0 TO 39: REM SPLIT=39, FULL=47
70 FOR X = 0 + N TO 79: & SCRN(
   X,Y,Z): & COLOR= Z: & PLOT
   X - N,Y: & COLOR= 0: & PLOT
   X,Y: NEXT X,Y
```

## CONVERT.LORES.2

This utility converts an existing lo-res picture to double lo-res. The picture will be duplicated by doubling each pixel, preserving the proportions of the picture. Here's the procedure:

1. Load DGR (unless it's already loaded):  
**BRUN DGR**
2. Select double lo-res graphics:  
**& GR**
3. Load the CONVERT.LORES.2 utility:  
**BLOAD CONVERT.LORES.2**
4. Load the lo-res picture to be converted. Substitute the appropriate name for picture:  
**BLOAD picture ,A\$4000**
5. Call the conversion routine:  
**CALL 768**

The picture is now converted. Save the new double lo-res picture using the LORES.LOAD.SAVE utility.

## GR.TO.DGR

This utility converts a normal Applesoft lo-res program so it works with double lo-res graphics. An ampersand will be placed in front of all regular lo-res commands ("GR" becomes "& GR", etc.). To convert a program, use the following procedure:

1. Load the Double Lo-Res Ampersand Inserter utility:  
**BLOAD GR.TO.DGR**
2. Load the program to be converted:  
**LOAD program**
3. Call the routine to insert the ampersands:  
**CALL 768**

Your program is now converted. Save it under a new name.

### NOTES

- An ampersand will be placed in front of all lo-res commands unless doing so would create an Applesoft line that is too long.
- If the SCRN command is used in the program, you will have to fix it yourself. It involves more than just inserting an ampersand before SCRN (see & SCRN in this manual, and SCRN in your Applesoft Manual).
- Lo-res and double lo-res color values are the same.

# LORES.LOAD.SAVE

This utility allows you to load and save double lo-res images *AND* 80-column text screens (double lo-res occupies the same memory locations as 80-column text). Use the following procedure directly from the keyboard or within a program:

## SAVING DOUBLE LO-RES GRAPHICS OR 80-COLUMN TEXT

1. Load DGR (unless it's already loaded):  
**BRUN DGR**
2. Select double lo-res graphics. Skip this step if the picture is already on the screen:  
**& GR**
3. Load the LORES.LOAD.SAVE utility:  
**BLOAD LORES.LOAD.SAVE**
4. Plot, print or load the desired image on the screen.
5. Poke the address of the temporary storage area (lo-byte, hi-byte format). Substitute 16384 for tempaddr if you want (see page 80):  
**POKE 0,tempaddr-INT(tempaddr/256)\*256**  
**POKE 1,INT(tempaddr/256)**
6. Call the routine to save the screen:  
**CALL 768**
7. Save the screen using tempaddr and a length of 1920:  
**BSAVE picture,A tempaddr,L1920**

## **LOADING DOUBLE LO-RES GRAPHICS OR 80-COLUMN TEXT**

1. Load DGR (unless it's already loaded):  
**BRUN DGR**
2. Select double lo-res graphics:  
**& GR**
3. Load the LORES.LOAD.SAVE utility:  
**BLOAD LORES.LOAD.SAVE**
4. Poke the address of the temporary storage area into memory. Use 16384 for tempaddr if you want (see page 80):  
**POKE 0,tempaddr-INT(tempaddr/256)\*256**  
**POKE 1,INT(tempaddr/256)**
5. Load the screen image:  
**BLOAD picture,A tempaddr**
6. Call the load routine:  
**CALL 768+3** (or **CALL 771**)



## APPENDIX

# Protecting the Hi-Res Screens

When Steve and Steve designed the Apple, they put the hi-res screens right in the middle of the memory area used by Applesoft programs (they had good reasons, but we don't want to hear about it). This set the stage for some serious problems (not to mention some screams in the night)—your programs can easily *overwrite* the memory area used by a hi-res picture. To get a feel for what's happening, take a look at the memory map on the color side of your Peeks & Pokes chart.

Applesoft programs usually start at address 2048 (\$800) and end somewhere above that. Variables are stored right above the program at a location called "Lomem", sort of attached to the program. If the size of a program plus its variables is large enough to occupy one or both hi-res pages *and* the program uses hi-res displays, then you've got trouble. Namely, little lines that start creeping into hi-res pictures. Or programs or variables that get zapped (erased) by an HGR command. Or programs that just plain crash.

If your program is small, no problem, but if it isn't, watch out! Here are a few techniques that will prevent hi-res overwriting.

## MOVE LOMEM TO RAISE YOUR VARIABLES

Making the command **LOMEM:16384** the first statement in a program will cause variables to be stored above hi-res page 1. **LOMEM:24576** will do the same above page 2.\* If your program (without its variables) is smaller than 6144 bytes,\*\* it will fit nicely between \$800 and \$2000, the area below hi-res.

\* Hi-res page 2 isn't used by double hi-res graphics; however, it may be needed as a temporary storage area (page 80) for the Beagle Graphics utility programs, or as a location to store a double hi-res character font.

\*\*To determine the length of a program, LOAD it, and then type:

```
PRINT (PEEK(175)+PEEK(176)*256)-(PEEK(103)+PEEK(104)*256)
```

Or look in the catalog and multiply the number of sectors times 256 and subtract 256 from the total (gives an approximate answer).

### **MOVE YOUR PROGRAM**

If you move your program above page 1, you will lose the 6144 bytes before page 1, but your programs can be up to 14K long. To change the program starting address above hi-res page 1 use the following pokes before loading your program:

**POKE 103,1: POKE 104,64: POKE 16384,0**

### **PUT A "HOLE" IN YOUR PROGRAM**

Go buy Beagle Bros' *Silicon Salad* disk. Among other useful utilities, you will find one called *Program Splitter*, which will divide a program into two sections, leaving a gap for the hi-res screen. This is more efficient because it makes use of the 6144 bytes below the hi-res pages as well as the space above.

### **MAKE ONE PROGRAM RUN THE NEXT**

Break your programs into different parts and have each part call the next part using the RUN or CHAIN command. See your DOS 3.3 or ProDOS instructions for more information.

# BEAGLE GRAPHICS INDEX



Apple IIe/IIc .....	6	DOUBLE.SCRUNCH utility ..	86
Auxiliary memory .....	55	Draw, Double Plot .....	16
Background color .....	24,37	& DRAW .....	41
Backups .....	4	Drive change .....	28
& BCOLOR .....	37	Dumping pictures .....	9
Box, Double Plot .....	15	Edit, Double Plot .....	17
& BOX .....	38	Ellipse .....	15,39,59
Brushes .....	22	Erase screen ...	25,40,45,46,66,67
Button, The .....	12-13	Fill, Double Plot .....	20
Catalog .....	5,26	& FILL .....	42
CHANGE.COLORS utility ...	81	Flip and flop .....	18
Character editor .....	74	FONT.EDITOR .....	74
Checkerboard fill .....	21	Fonts .....	23,26
Circle .....	15,39,59	Foreground color .....	24,44
& CIRCLE .....	39	Formatting blank disks .....	4
Clear screen ...	25,40,45,46,66,67	Free cash .....	90
& CLEAR .....	40	& GOTO .....	43
Colors, Double Plot .....	24	& GR .....	66
CONVERT.HIRES.1 utility ...	82	& GR2 .....	67
CONVERT.HIRES.2 utility ...	83	GR.TO.DGR utility .....	93
CONVERT.LORES.1 utility ...	91	Graphics tablet .....	12
CONVERT.LORES.2 utility ...	92	Hardware requirements .....	6
& COLOR .....	65	& HCOLOR .....	44
Copy, Double Plot .....	18	& HGR .....	45
CUT.AND.PASTE utility .....	84	& HGR2 .....	46
Cut and Paste, Double Plot ...	17	HGR.TO.DHGR utility .....	88
DGR .....	64	& HLIN .....	68
DHGR .....	34	& H PLOT .....	47
Disk commands .....	26	& HSCRN .....	48
DOS 3.3 .....	4	Icon switching .....	29

Initializing blank disks .....	4	ProDOS .....	4
Invert, Double Plot .....	18	Program converter .....	88,93
Joystick, paddles .....	13	Protecting hi-res .....	96
Keyboard .....	13	Quick mode .....	12,13,29
Line, Double Plot .....	16	Quit .....	25
Load, Double Plot .....	27	Resolution .....	8,32,64
& LOAD .....	49	RGB monitors .....	6,7
Lomem .....	96	& ROT .....	54
Lo-res commands .....	64	Save, Double Plot .....	28
LORES.LOAD.SAVE utility ...	94	& SAVE .....	55
Main menu .....	14	& SCALE .....	56
Modes .....	7	& SCRN .....	70
Mode, Double Plot .....	24	Set color .....	24
& MODE .....	50	Shape tables .....	41,54,56,60
Monitors .....	6	Silicon Salad .....	97
Mouse .....	12	SLIDE.SHOW .....	76
Move, Double Plot .....	18	Stripe fill .....	21
& NORMAL .....	51	Tempaddr .....	80
Packed pictures .....	86	Text, Double Plot .....	23
PAGE.2 utility .....	89	& TEXT .....	57,71
Paint, Double Plot .....	22	Triple-Dump .....	9
Patterns .....	20	Trouble shooting .....	33,96
Picture converter .....	82,83,91,92	Typing .....	23,43,52,62
Pixels .....	8	Utilities .....	79
& PLOT .....	69	& VLIN .....	72
Pointing devices .....	12	& XBOX .....	58
Precision mode .....	12,13,29	& XCIRCLE .....	59
& PRINT .....	52	& XDRAW .....	60
Printing, hi-res .....	23,43,52,62	& XPLOT .....	61
Printing pictures .....	9	& XPRINT .....	62

### **Disclaimer of All Warranties and Liabilities**

Even though the software described in this manual has been tested and reviewed, neither Beagle Bros nor its software suppliers make any warranty or representation, either express or implied, with respect to this manual, the software and/or the diskette; their quality, performance, merchantability, or fitness for any particular purpose. As a result, the diskette, software and manual are sold "as is," and you, the purchaser, are assuming the entire risk as to their quality and performance. In no event will Beagle Bros or its software suppliers be liable for direct, indirect, incidental, or consequential damages resulting from any defect in the diskette, software, or manual, even if they have been advised of the possibility of such damages. In particular, they shall have no liability for any programs or data stored in or used with Beagle Bros products, including the costs of recovering or reproducing these programs or data. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

### **DOS 3.3 and ProDOS™**

This product includes software, DOS 3.3 and ProDOS, licensed from Apple Computer, Inc. Apple Computer, Inc. makes no warranties, either express or implied, regarding the enclosed computer software package, its merchantability or its fitness for any particular purpose. Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you.

**BEAGLE GRAPHICS™**, Copyright © 1984, Mark Simonsen  
ISBN 0-917085-02-7

"APPLE" is a registered trademark of Apple Computer, Inc.

Published by **BEAGLE BROS MICRO SOFTWARE, INC.**  
3990 Old Town Avenue, San Diego, California 92110