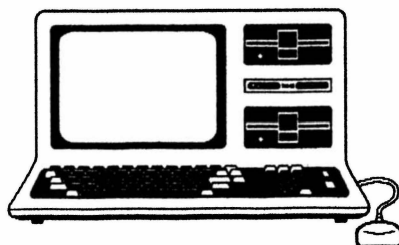


SCOTT MCBURNEY'S
MDRAW II

**HI-RESOLUTION DRAWING FOR
THE MODEL 4 TRS-80 WITH
HI-RES BOARD, GBASIC, AND LS-DOS 6.3
WITH FULL SUPPORT FOR
MOUSE INPUT AND DOTWRITER FONTS**

Documentation by
Gary W. Shanafelt
DEPT. OF HISTORY
McMURRY UNIVERSITY
ABILENE, TX 79697



This manual rendered on a Model 4 TRS-80 running Prosoft's Allwrite word processor with the *Computer News 80* DeskJet Utility Pack, using 18 and 30 point Broadway, 18 point Parisian, and 12 point Kabel typefaces on the title page and 10 point Kabel in the text. Graphic designs created or edited with MDRAW II and printed with HPPRINT/BAS. Version 4.x, June 1992.

Introduction

Scott McBurney first wrote this program as a basic drawing utility for his Model 4 hi-resolution board. He called it DRAW/BAS. Though he said it was simply a stopgap for MicroLabs' Pro-Draw, which he couldn't afford, it became part of everyone's hi-res library because it included a routine to print Dotwriter letters on the screen. It had something no other program had. George Madison liked it so much, in fact, that he rewrote the Dotwriter routines and ported them over to TRSDRAW/BAS, another hi-res drawing program. Then, a few years later, Scott wrote a revised version called MDRAW/BAS. It was basically DRAW but with the addition of a driver to support a Microsoft serial mouse: hence the "M" in the name, for "mouse." This was again a first -- no other TRS-80 program would work with a serial mouse. And when the MicroLabs mouse hardware interface was sold out, MDRAW/BAS became the ONLY way to use a mouse with the hi-res board.

In the last year, a number of TRS-80 mouse drivers have become available, all using the same interface for the Model 4. They were inspired by (you guessed it) the original mouse driver that Scott wrote for MDRAW. It thus seems like the time to upgrade MDRAW to work with them. David Goben established the new TRS-80 mouse standard and rewrote the interfacing in the program to work with it. Matthew Reed wrote the actual mouse drivers provided with the program. Yours truly worked over much of the rest of the code, adding or changing various features. Finally, to enable the program to use both the normal /HR format hi-res files and the compressed /CHR format, the program now utilizes two utilities written by Mel Patrick (with minor modification by me), BCHRLOAD/CMD and BCHRSAVE/CMD. This really has been something of a cooperative effort, but it all happened because Scott McBurney wrote such an original series of programs to start with. Despite all the changes, MDRAW II is still Scott's program.

What You Need

First, you need a Model 4 TRS-80 computer with a hi-res board. Second, you need LS-DOS 6.3.x, for the program uses the USR11 call introduced with LS-DOS BASIC to access the DOS SVC's (and thus the mouse driver). It won't work with TRSDOS 6.2. Third, you need MicroLabs' GBASIC, for it is programmed in that language. If you have Radio Shack's rather than MicroLabs' board, you need the special GBASIC that MicroLabs makes for it, or else the text/graphics overlay routines won't work properly. Fourth, you need one of the new mouse drivers, either that by David Goben or those by Matthew Reed. The latter are included with this package. Though the point of all this is to use a mouse, the program will actually run without one if you have all the other things -- you can use the arrow keys instead. (You MUST load a mouse driver even if you don't have a mouse, by the way, because of the way the program interfaces with the DOS SVC's). Make sure you get a serial mouse. This one is so called because it plugs into the serial (RS232) port of your computer with no need for any additional hardware. You should make sure you don't get a bus mouse, which requires a separate interface board.

Running MDRAW II

To run the program, you have to first install the proper driver for your mouse, then load GBASIC, and finally run MDRAW/BAS. A quick loader is provided to automate the process. If you have a 3-button mouse, type RMDRAW3 and hit the <ENTER> key. RMDRAW3/CMD loads MMOUSE3/CMD, then GBASIC, and finally runs the program. RMDRAW2 does the same thing for a 2-button mouse. (If you have the Goben mouse driver, use GMDRAW3 or GMDRAW2 respectively).

Once you decide which driver you will be using, you should rename the quick loader to simply MDRAW/CMD. In the future, you'll load the program simply by typing

MDRAW <ENTER>

Mdraw II -- Page 2

After the title, which will display for a time while the data file MDRAW/DAT is loaded, you'll be prompted to clear the graphics screen. If you have 128k, all three screen areas will be cleared, so this will take a moment. Any other key but <Y> -- or the left mouse button -- takes you directly into the program, with the current screens unchanged. You'll see a small arrow -- the MDRAW cursor or pointer -- and you're ready to begin.

Now, a brief note about terms. Generally, you'll be SETting or CLEARing dots on the graphics screen. To SET a dot means to turn it on. CLEARing a dot means turning it off. The latter process is sometimes called RESETing; I'm using the terminology CLEAR because it is clearer (pun?) for most people as well as consistent with the usage in the GBASIC manual. Occasionally, you'll have a third option, to NEGATE the dots. This is actually a fancy term for complementing or reversing the dots, but <C> couldn't be used as a command since it was already taken for "Circle," and in any case you're probably used to the term "negate" if you've ever used MicroLabs' Pro-Draw. When you negate an area, everything that was on is turned off, and everything that was off is turned on. You might be wondering why you'd want to CLEAR dots. There are two reasons. First, this is one way of erasing things from the normal white-on-black screen. Second, if you use a black-on-white screen (which many CompuServe GIF files use) -- that is, one with a white background instead of a black one -- then CLEARing dots actually creates a picture for you rather than removing it. In most functions, the left mouse button SETs, and the right one CLEARs.



The Commands

From the main section, you have a number of options.

THE MOUSE -- The cursor can be moved by the mouse. Just move the mouse and the pointer on the screen should move with it. If you hold down the left button, a line of dots will be SET. If you hold down the right button, a line of dots will be CLEARED. This allows you to do freehand drawing with either a white-on-black screen or a black-on-white one. Pressing the middle button (or both buttons at once on a 2-button mouse) duplicates the <ENTER> key and returns to the default conditions for a number of commands. The mouse buttons also function within many of the commands below. In general, you'll use the left button if you're editing a white-on-black screen, or the right button for a black-on-white one. Thus, a function like drawing a box or a line will have the left button setting the dots and the right one clearing them. Similarly, a function to remove a line (like Erase or Undo) will have the left button clearing and the right button setting. The settings should become clear through actual usage.

THE ARROWS -- These also move the cursor. The arrows by themselves move the cursor one dot in each direction. This is slow, but good for detail work. <SHIFT><ARROW> together moves the cursor several dots in each direction. Finally, <CLEAR><ARROW> moves all the way to the edge of the screen in whatever direction the arrow indicates. If you have the mouse at a slow setting, this is actually a faster way to move the cursor across the screen than with the mouse itself.

THE FUNCTION KEYS -- These provide further cursor options. <F1> toggles automatic setting on and off. It works only if you are using the arrow keys to move the cursor. If automatic setting is on, a line is drawn as the cursor is moved. <F3> does the same thing except that it clears lines. Hitting <F1> or <F3> again toggles the function off. The keys have no effect, by the way, when you're moving the cursor with the mouse. They are included for people without mice, to make drawing with the arrows alone a little easier. Finally, <F2> puts the cursor in the middle of the screen.

ENTER -- The <ENTER> key generally aborts functions. Here, it turns off spray painting, jagged line mode, automatic setting and clearing (F1 and F3), or viewing of the X,Y coordinates if any of them is on; returns the mouse sensitivity to 1; and sets the line width to 1. These are all the default conditions when you start the program.

THE SPACE BAR -- <SPACE> generally completes functions. Here, it negates (or complements) whatever dot the cursor is on. That means that if the dot is clear, it is set or turned on; or, if it is already set, then it is cleared or reset or turned off. The number of dots affected depends on what kind of line width was selected or if the spray paint or jagged line options are active. <SPACE> also turns off <F1> or <F3> if either is on.

1-4 -- Hitting any one of these keys varies the line width from 1 to 4 dots. 1 is the default. The new width is in effect when drawing circles, boxes, and lines as well as with freehand drawing.

<SHIFT><CLEAR> -- Clears the graphics screen. You'll be prompted Yes or No.

A -- Adjust mouse sensitivity: 1 is the default (and slowest) setting; 4 is the max. Either <ENTER> or <SPACE> or clicking a mouse button returns you to the graphics screen without making any change. 1 or 2 seem to be best for freehand drawing. This is because MDRAW uses the @LINE command in drawing mode, and at the faster speeds the mouse moves faster than BASIC can interpret it, so that you get jagged lines where you want curves.

B -- Draw a box: Put the pointer where you want one of the corners of your box. Then hit and a box will appear. Use the mouse or the arrows to move it until it's the size you want. The left mouse button or the <S> key sets the box; the right button, the <C> key, or the <R> key clears (resets) the box; the <N> key negates (complements) the box; and the <J> key creates a box with a jagged (dotted) line. <SPACE> duplicates the <S> key and left button. If you change your mind and decide you don't want a box after all, hit the <ENTER> key or the middle mouse button.

C -- Draw a circle: Put the graphics pointer where you want your circle to be centered. Hitting <C> causes a cross to appear on the screen, indicating the diameter of your circle. Use the right/left arrows or mouse to get the size you want. The left mouse button or the <S> key or <SPACE> set the circle; the right button, <C>, or <R> clears it; <ENTER> aborts.

D -- Disk access: Pressing <D> gives you a menu of disk choices, namely to load, save, or kill a file, as well as seeing a disk directory. MDRAW will handle both 19.5K /HR files and Mel Patrick's compressed format /CHR files. You'll need the utilities BCHRLOAD/CMD and BCHRSAVE/CMD online in order to process the /CHR files. If they aren't there, you'll get a File Not Found error when you try to do something with one of these type graphics files.

If you're loading or saving, you'll get two prompts. First, you'll be prompted for the file name. You do NOT have to type the extension (HR or CHR); the program will add it automatically if none is specified. Then, you'll be prompted for type of file (HR or CHR). HR is the default if you just hit <ENTER>. If you want CHR to be the default, the lines to change are 390 and 430. If you want to kill a file, the extension "/HR" will be added automatically if you don't type an extension yourself. If you want to change the default extension to "/CHR", the line to change is 2790. If you want to load a file with no extension, type "/" after the file name, i.e., "FILE".

If you're not familiar with the /CHR format, note the included demo file, ICONS/CHR. It is 15k in length; the same file in /HR format would be 19.5K. You'll see the greatest size difference in files which have a lot of clear area, for this is where the compression algorithm is most effective.

E -- Erase: A small eraser will appear when you hit <E>. To erase, use the left button, <C>, or <R> to clear; the right button or <S> to set. (On a normal white-on-black screen, clearing the dots removes a design, while setting the dots on a black-on-white screen removes a design). <SPACE> also clears. Leave the function by hitting <ENTER> or the middle mouse button. This function is good for medium-sized erasing.

F -- Fill: Fill an area with any pattern from 0 to 255. 255 is solid, 192 is vertical lines, 127 is horizontal lines, etc. Either <ENTER> or <SPACE> aborts. Make sure the area you want to fill is surrounded by a solid line; otherwise, the fill pattern will "leak" out and cover the entire screen!

G -- Get a block of graphics: This command is used in conjunction with the Put command below to move blocks of graphics on the screen. Place the cursor at the corner of the block you want to store and press <G>. A box will appear on the screen. You move the arrows or mouse to include everything you want, then press either the left or the right mouse button or <SPACE> or <G> to get it. There should be enough memory to get about a forth of the screen. If you try to get too much, you'll get an Out Of Memory message and the command will automatically abort. (If you find this happening a lot, there are two things you can do: try using the Reposition command to move the whole screen, or enter GBASIC with the (F=1) parameter). <ENTER> or the middle button aborts. While the computer is storing the block, both the cursor and the box will disappear. The process can take some time for a large block, so be patient.

The above allows you to make a copy of a block to duplicate somewhere else on the screen (or on the alternate screen with the <@> command if you have a 128k machine). If you want to move a block, that is, erase the original occurrence and not just duplicate it, you can do so with two further commands once you've defined your block with the original <G> command. <C> or <R> clears (or resets) the area of the block after the data has been stored; <S> sets it. Use Clear if you're working with a white-on-black screen; Set for black-on-white. Again, these commands store the block just like hitting the mouse buttons or <SPACE>; the only difference is that after doing so, they erase the original location that you boxed.

I -- Invert the screen: Flips the screen upside down.

J -- Jagged line mode: Draws a jagged (dotted) line. It should be obvious to the perspicacious user that this command would have been <D> except that that letter was already being used for "Disk access"; "jagged" was the closest synonym I could think of for "dot" or "dash" without preempting another letter already in use. This feature is

Mdraw II -- Page 5

toggled on and off; turning jagged lines on automatically turns spray paint off, if the latter is on.

L -- Draw a line: Put the cursor at the start of the line, then press <L>. You use the mouse or the arrows to position the line where you want it. The left button or <S> sets it, the right button or <C> or <R> clears it, <N> negates it, and <J> creates a jagged (dotted) line. <SPACE> sets while <ENTER> aborts.

M -- Mirror image of the screen: Flips the screen from right to left.

N -- Negate the screen: Negates or complements every dot on the screen. If you started out with a white-on-black screen, negating it gives you a black-on-white one, and vice-versa.

O -- Output the screen to a printer: GBASIC supports ten printers through its @LPRINT command. Right now, the program is set to @LPRINT(1), which is for a standard Epson printer. If you want to use a different printer, simply go to line 1390 and change that number to whatever printer among the ten that you want; then save the program back to disk. If you have a laser or DeskJet printer -- I have a DeskJet -- you can use either my HPPRINT/BAS program or David Goblen's laser printing utility to print your screen once you've gotten it the way you want it with MDRAW. In the case of the former, change line 1390 to **IF A\$="O"THEN RUN "HPPRINT/BAS"**. (The latest version of HPPRINT/BAS, by the way, has a new command to run MDRAW, so that the two programs easily integrate together).

P -- Put a block of graphics: This command works in tandem with the Get command above. Once you've gotten a block, you press <P> to put it anywhere on the screen. A box the size of the original box you defined with the Get command will appear. This shows you where the block will be placed. Use the arrows or mouse to move the block to wherever you want it. When you have it right, you have several options. The left button or <O> will cause the block to Overlay or replace what was already there on the screen. The right button or <M> will Merge the block with whatever is already there, so that the new image overlaps the first. Finally, you have the <T> option, for "test." <T> causes the block to be XORed to the screen, so that you can test where, precisely, it is going to be placed. Hitting any key then removes the image and restores the box, which can be repositioned as necessary before you Put the image permanently. This works best with a normal white-on-black screen, as you'll find out if you try to XOR a screen with a lot of white. <SPACE> duplicates the overlay function, while <ENTER> aborts.

The Put command can be maximized in conjunction with several other commands. If you want to replicate a block either upside down or backwards from the original source, get the block, then use the <I> or <M> commands to change the orientation of the screen, put the block, then reorient the screen... and voila, you've flipped or reversed the original block. You can do the same thing with the <N> command to create blocks that are negations or complements of the original screen.

R -- Reposition screen: This command allows you to shift the whole screen in any direction. When you hit "R", the cursor disappears and you will be in Reposition mode. Each arrow moves the screen one dot in the direction specified, with the line of dots that scroll off one edge being put back on the opposite side. You get the same effect by moving the mouse, but the arrows seem more precise. If you're in a hurry (this IS slow) use the arrows with the SHIFT key. This will scroll the screen about 1/4 inch at a time. This command is useful for centering images that are smaller than the whole screen, or for cropping sections of a larger image. To get back to normal mode, hit <ENTER>, <SPACE>, or any mouse button. You'll know you're back when the cursor reappears.

S -- Spray paint mode: Instead of a single dot or line, spray paint mode allows you to place a cluster of dots on the screen. By repeating the clusters, you can get a spray paint pattern for shading images or creating special effects. In spray paint mode, the left mouse button sets clusters of dots when pressed rather than setting a line; the right button clears clusters of dots instead of clearing a line. The clusters are affected by the line width command, and work best when the width is at its default of 1. If you try spray painting with a width of 4, you get blobs rather than dot clusters... This command is toggled off and on; hitting <S> a second time returns you to normal line mode. It also automatically toggles jagged line mode off, if the latter was earlier activated.

T -- Text entry: There are a number of options here. If you chose normal text entry, you can input text in any of the

Mdraw II -- Page 6

four modes supported by the hi-res board and GBASIC. When you choose which mode you want, a small box the size of the letters will appear. Use the arrows or mouse to position the box, and begin typing. In addition to the normal keyboard letters, you can also input most of the TRS-80 special characters. You do this by holding down the <CLEAR> key and then pressing some other key. <CLEAR><A>, for example, will print a heart on the screen. Finally, you can edit normal text using the <SHIFT><ARROW> combinations, for they will move the box the exact amount to reposition over a letter. To erase a letter, use <SHIFT><ARROW> to reposition over the letter and then retype the same letter. (If you try to reposition with the mouse or an arrow by itself, you'll discover it's just about impossible without being off a dot or so). Hitting <ENTER> or any mouse button exits text mode and returns you to the normal screen.

The unique feature of the original MDRAW was its ability to use Prosoft's 8-bit Dotwriter fonts to print fancy letters on the screen. That feature is still here (naturally). If you're not sure what fonts you have on a disk, you have the option of doing a directory of all the Dotwriter files. Note: The original Dotwriter files are designated by the suffix /PR (normal), /PRI (Italic), or /PRB (banner or sideways letters). Some public domain files use a suffix /DOT. If you have one of these, you should rename the suffix to /PR or else this directory feature won't recognize them. When your font is online, select <D> for Dotwriter from the main text menu, then enter the name of the font you want. Unless you specify an extension, the suffix "/PR" will be added automatically (that is, if you want the font BNOST1/PR, included with this package, you need only type "BNOST1"; the program will add the "/PR" for you). You'll then be prompted for the spacing in dots you want between the letters. The default is 4, and this is the number you'll get if you hit either <SPACE> or <ENTER> or click the mouse at this point. Then, your disk drives will whir and you'll be returned to the graphics screen with a box the size of the maximum letter in the font telling you where the first letter will be placed. Use the mouse or the arrows to reposition at any time. Type in what you want. Remember that not all Dotwriter fonts have all the letters or symbols you see on your keyboard, and some have no lowercase letters. There are also a few which have non-standard headers, so that they will not print correctly on the screen. (Some of these can be "fixed" by running them through the proportionalizing utility in the original Dotwriter package). Since most Dotwriter fonts are proportionally spaced, <SHIFT><ARROW> won't give you exact letter positioning as was the case with normal, fixed-pitch text. When you're done, hit <ENTER> or click either mouse button.

U -- Undo/erase: Put the pointer at any corner of the area you want to undo, and hit <U>. Once again, the box appears, and grows as you move the mouse or arrows. Box the area to be undone. Hitting <C>, <R>, or the left button clears everything within the boxed area; hitting <S> or the right button sets everything. This is good for deleting large areas on the screen, not detail work. Like the Erase command above, the button assignments are the reverse of the usual. You'll use the clear option with a normal white-on-black screen, but the set option if you're editing a black-on-white screen. <SPACE> clears, <ENTER> aborts.

V -- View X,Y coordinates: The graphics screen measures 0-639 dots across, 0-239 dots down. Positioning is done by locating a point on the horizontal (X) axis and on the vertical (Y) axis. This command shows the current X,Y position of the cursor in the lower right hand corner of the screen. It can be helpful in positioning blocks of memory (see the Get command above). View mode is toggled on and off; normally it is off so as not to block your view of the graphics screen. You should probably leave it off if you're doing freehand drawing, for you can get smoother lines when the program doesn't have to jump to a subroutine to reprint the coordinates every time the mouse is moved.

W -- Whoops: If you have a 128k machine, this gives you a backup screen if you screw up. Hitting <W> gives you a menu with 2 choices. <S> saves the current screen to memory, while <L> loads whatever is in the memory onto the screen. <ENTER>, <SPACE>, or any mouse button aborts without doing anything. If you <L>oad something onto the screen and then decide you want the original screen back, hit <W> and then hit <L> again. When <L> is invoked and the screen in memory is loaded into view, the current screen is transferred to memory as a safety precaution in case you don't like its replacement. The Whoops command is intended to help you with major editing. If you're about to make changes to an image and you're not sure you're going to like them, hit <W>, then <S> to save the current screen to memory. Make your revisions. If the changes don't look right (WHOOPS!) hit <W> again and press <L> to load the original screen from memory back into view. If you then decide you like them after all, another <W> followed by another <L> will bring them back. (This really isn't as confusing as it sounds).

Mdraw II -- Page 7

<W> uses bank 2 of the Model 4 extended memory; if that bank isn't available, it will abort with a beep. It is similar to the <@> function below except that the two screens overlay each other rather than being toggled. Since the <W> memory buffer is separate from the <@> one, you can use both functions at the same time with a total of three screens (one in view, the other two in the two respective memory buffers).

X -- Exit to DOS: Leave the program. "Yes" is the default. <N>, <ENTER>, or the middle mouse button return you to the program; any other key exits back to LS-DOS ready.

Z -- Zoom: Magnifies a portion of the screen for detail work. In zoom mode, you have access to most of the same keys available in normal drawing mode. Use the arrows to move, or the arrows with <SHIFT> or <CLEAR> to move several pixels at a time. <SPACE> toggles pixels on and off. <F1> and <F3> will automatically set or clear all pixels when you move the cursor; hit them again to toggle them off (hitting <SPACE> automatically turns them off). The mouse is also available for input, but the arrow keys are more precise. <ENTER> (or the middle mouse button) returns to the regular screen with your changes; <SHIFT><CLEAR> returns without the changes.

@ -- Toggle two screens: If you have a 128k machine, <@> toggles between two separate graphics screens. Note that the memory used for swapping the screens is separate from that used for the <W> function above, so that neither screen swapping function preempts the other. Load a file into view on the hi-res display, then hit <@> to toggle with its alternate screen. You can now load in a second file, hitting <@> at need to switch from one to the other. You can use the Get and Put functions to move blocks of graphics from one screen to the other. This is useful if you have files of clip art; you can put the art in one screen and then transport it to your working screen at will. The sample file, ICONS/CHR, provides some common images. This function is similar to the <W> function above except that the two screens are always kept separate, whereas with Whoops they load over each other. <@> uses bank 1 of the Model 4 extended memory; if that bank isn't available, the command will abort with a beep.

? or / -- Prints a brief command summary on the screen. Hit any key or mouse button to return to the graphics screen. If you want to print the file for quick reference, press <?> to display it on the screen and then press <CTRL><*>. This is the DOS screen dump command, and it will send the whole command screen to your printer without further ado (assuming the printer is turned on... otherwise, you'll get an I/O error message).



Using MDRAW II with Double Duty

MDRAW/BAS works fine with Double Duty if you have a 128K machine (though you lose the <@> and <W> functions). Be careful, however, always to install the mouse driver BEFORE installing Double Duty. Since the driver adds a custom SVC to the DOS, you can't have it present in one bank but not the other unless you want the risk of crashing the system. Double Duty is especially nice if you want to run another graphics program simultaneously with MDRAW, to do things which MDRAW does not support. (No program has everything: you may want to use MicroLabs' Pro-Draw, for example, to draw arcs or resize graphics blocks). If you run MDRAW with another graphics program, the MDRAW cursor will be frozen on the screen while you're switched out of it, so you should put it out of the way before switching. And if you run MDRAW with a non-graphics program (I'm running it in one bank while I type this with Allwrite in the other, switching back and forth as necessary to check each command), you should turn off the graphics screen before switching banks. This is done by invoking any MDRAW command that replaces the screen with text. When you return, just hit <ENTER> to abort the command and return to the screen.

What If Something Goes Wrong?

If strange things seem to be happening, the first thing to try is to hit <ENTER> from the main section of the program. That will return the mouse sensitivity, line width, and other values to their default conditions (this is assuming you initiated some special function without realizing it, or else forgot about it). If you get the "Wrong DAT file" message at the start of the program, you forgot to copy the MDRAW/DAT file that comes with the program to your working disk. Transfer the file and try again.

I've occasionally had problems with the graphics cursor either not moving at all, or drawing a line even when no mouse buttons were pressed. This seems to be caused by a faulty mouse/computer connection. My mouse has several adapter plugs on it to make the 9-pin mouse plug connect with my 25-pin RS232 cable plug. If any of them is loose, the mouse does weird things -- or nothing at all. You can also get in trouble if there is a crimp in the mouse cord. Try unplugging the mouse to clear whatever values are stored there, and then re-run MDRAW. Finally, make sure you're using the right driver with the right mouse. A 2-button driver with a 3-button mouse, or vice-versa, is certain to screw up. Some 3-button mice have a hardware switch to select for 2-button emulation; make sure the switch is correctly set for the emulation you want. If worse comes to worst, remember that you can reboot your whole computer without affecting what's stored on the graphics board; just don't clear the screen when you re-enter MDRAW, and you'll find everything as you left it.

Distribution, Etc.

This program is freeware, which means you can give it to whomever wants it free of charge. If you can't figure something out or find a bug that needs to be fixed, please let us know. Shanafelt's address is on the title page of this manual. McBurney can be reached via EMAIL on GENIE through ID S.MCBURNEY or at

13700 West Smith Road
Manhattan, IL 60442



README/NOW -- Quick documentation for the MDRAW II package
Gary W. Shanafelt
June 1992

Sooner or later, you need to read the full documentation for MDRAW II. But if you want to get started right away, this will tell you how. If you already have version 2.x of the program, refer to CHANGES/DOC to find out what has been changed...

In this package, you should find the following files. These should be online when you run the program:

MDRAW/BAS -- The main program
MDRAW/HLP -- Command/help screen for MDRAW/BAS
MDRAW/DAT -- Data file for MDRAW/BAS
MMOUSE2/CMD -- Matthew Reed's 2-button mouse driver
MMOUSE3/CMD -- Matthew Reed's 3-button mouse driver
BCHRLOAD/CMD -- Mel Patrick's utility to load CHR files
BCHRSAVE/CMD -- Mel Patrick's utility to save CHR files

Plus one of the following, renamed MDRAW/CMD:

RMDRAW3/CMD -- Quick loader for Reed 3-button mouse driver
RMDRAW2/CMD -- Quick loader for Reed 2-button mouse driver
GMDRAW3/CMD -- Quick loader for Goben mouse driver set for 3 buttons
GMDRAW2/CMD -- Quick loader for Goben mouse driver set for 2 buttons

Of course, BASIC and GBASIC must also be online.

If you have a HP LaserJet or DeskJet printer, to print hi-res screens you'll need

HPPRINT/BAS -- HP hi-res screen dump utility

The following sample picture and Dotwriter files are also on the disk:

ICONS/CHR -- Various icons in CHR format
ICONS2/CHR -- More icons in CHR format
TRS80/CHR -- A Model 4 in CHR format
DRAWFILL/HR -- Reference table of all hi-res fill patterns
BONOST1/PR -- A public domain Dotwriter font by Mike Pew
BW/PR -- A public domain Dotwriter font by Mike Pew

The DOC files are all in ASCII format; print them with your word processor for future reference.

MDRAW/DOC -- Documentation for MDRAW/BAS
MMOUSE2/DOC -- Documentation for MMOUSE2/CMD
MMOUSE3/DOC -- Documentation for MMOUSE3/CMD
FONTS/DOC -- Documentation for the two Dotwriter fonts
HPPRINT/DOC -- Documentation for HPPRINT/BAS

CHANGES/DOC -- How ver. 4 (this one) differs from ver. 2
README/NOW -- This file

To run the program as-is, you need a Model 4 with a hi-res board, a serial mouse, and GBASIC. You also need LSDOS 6.3.x; TRSDOS 6.2 won't work. MDRAW II also has several features that take advantage of the extra banks of memory if you have a 128k machine. If you don't have a mouse, you can use the arrow keys to move.

If you have a 3-button mouse, type the following from LSDOS Ready:

RMDRAW3 <ENTER>

If you have a 2-button mouse, type:

RMDRAW2 <ENTER>

That should do it. RMDRAW3 or RMDRAW2 will automatically load the necessary Reed mouse driver, GBASIC, and MDRAW/BAS.

If you're using the Goben driver (which you have to order on your own), type either GMDRAW3 or GMDRAW2 and hit the <ENTER> key.

From within MDRAW, either "?" or "/" will call up a help screen of all the major commands (MDRAW/HLP).

At some point, you should for your own convenience rename whichever quick loader you decide to use to MDRAW/CMD. If something doesn't seem to be working, read the main documentation files, particularly MDRAW/DOC. You'll have to read these anyway, sooner or later, if you really want to understand how the program works.