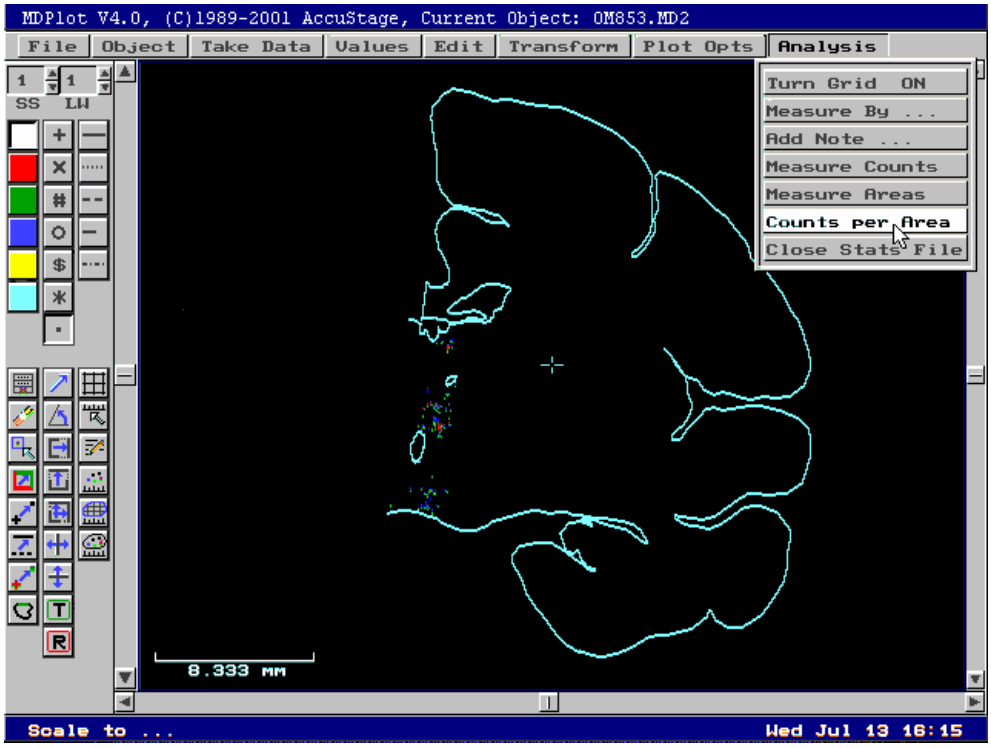
MDPlot Reference Manual

Data Acquisition and Plotting Software for the MD3 Microscope Digitizer

Version 4.0
January 2001



AccuStage
1000 Ingerson Road
Shoreview, MN 55126-8146, USA
+1 (651) 494-9791
http://www.accustage.com

Warranties and Limitations:

The software is provided on an "as is" basis only. AccuStage makes no representations, express or implied. AccuStage MAKES NO REPRESENTATIONS OR WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that the Software will not infringe any privately held patent, copyright, or trademark. The entire risk as to the results and the performance of this software is assumed by the User.

Information in this document is subject to change without notice and does not represent a commitment on the part of AccuStage. Not all copyrights pertain to all products.

U.S. Government Restricted Rights Legend:

Use, duplication or disclosure by the Government is subject to restrictions as set forth in FAR 52.227 -19(c)(2) or subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause as DFARS 252.227 - 7013 and/or similar or successor clauses in the FAR, or the DOD or NASA FAR supplement. Unpublished -- rights reserved under the Copyright Laws of the United States. Contractor/Manufacturer is AccuStage, 1000 Ingerson Road, Shoreview, MN 55126-8146.

Trademarks:

Microsoft, Windows, MSDOS, Windows 2000 and Windows NT are registered trademarks of Microsoft Corp.

MDPlot and **MD3 Microscope Digitizer** are trademarks of AccuStage.

All other copyrights, trademarks and registered trademarks are the property of their respective holders.

Acknowledgements:

Some portions of the software have been contributed by Greenleaf Software, Inc., 16479 Dallas Parkway, Suite 570, Dallas TX 75248, and are included in the binary portions of the code.

Table of Contents

Chapter 1: Introduction

This manual explains how to install and use the **MDPlot** software for the MD3 Microscope Digitizer. **MDPlot** allows you to conveniently take Digitizer data on a PC type computer. The computer must be equipped with an Enhanced Graphics Adapter (EGA), a Virtual Graphics Array (VGA), or a Super Virtual Graphics Array (SVGA) an EGA, VGA or SVGA color monitor, and a mouse. The software supports output of graphics data to a digital plotter, such as the Hewlett-Packard 7475A or a Houston Instrument DMP Series plotter, or to an HPGL/2 compatible color printer, such as the Hewlett-Packard DeskJet 1200C and 1600C. Digital plotters require that your computer have a free serial port, whereas an HPGL/2 compatible printer is normally connected to the standard printer parallel port. Graphics data can also be saved in HPGL/2 format disk files, which can be imported into many popular computer-aided-design and word processing programs.

Overview of this Manual

Chapter 1: Introduction, page 1, provides an overview of this manual, briefly introduces the main program, **MDPlot**, and the conventions used throughout the manual.

Chapter 2: Installing the System, page 3, provides instructions for connecting the MD3 to your computer, and for software installation using the interactive program, **install.exe**.

Chapter 3: Program Operation, page 7, discusses the use of the pull down menus, text entry in directory and file listing widgets, and the function of various special keys. Use of the Resource File, *$mdprc.ini*, in configuration of the software is described.

Chapter 4: Program Description, page 12, describes the logical groupings of functions in **MDPlot** and use of the menus for each.

Chapter 5: MDPlot Utilities, page 24, describes use of the diagnostic program **ptest.exe** that can be used to test serial output to the plotter, and the program **plotfile.exe** for output of files in HPGL/2 format to your plotter or printer.

Appendix A: MDPlot Source Code, page 25, gives an overview of the C language source code modules, header files and function library that are supplied with the software.

Appendix B: MDPlot Resource File, page 27, lists the format of the **MDPlot** Resource File, *$mdprc.ini*, a text file used to set program operation.

Appendix C: Data File Format, page 33, details the format of **MDPlot** data files.

Appendix D: Plotter/Printer Cable Configuration, page 37, describes the different configurations of serial and parallel cables required to connect a PC to a digital plotter or to an HPGL/2 compatible color printer.

Throughout this manual it is assumed that you already have some experience with the Microsoft Windows and DOS operating systems and use of your PC. You should know how to start up the computer, how to navigate around the directory tree, list contents of directories and manipulate files. Skill with a word processor is desirable but not necessary.

Conventions

Throughout this manual the following conventions are used:

- In text descriptions, filenames and DOS commands are in italics and the **MDPlot** program name and associated **MDPlot** utility function names are in **bold face** type. Literals (things that you should type exactly as they are shown) are in Arial (Helvetica) font.
- Text descriptions of menus use underlined Arial font for the menu name and when describing the names of menu items.
- In command syntax and examples, square brackets [] surrounding an argument indicate that the argument is optional. Variable parameters and file names are in *italics*. You replace these variables with appropriate strings or names of your own.

CHAPTER 2: INSTALLING THE SYSTEM

Installing the MD3

Connections to the Computer

The MD3 Microscope Digitizer communicates with your PC via one of the computer's serial ports COM1 through COM4. (Note: Not all PC's have COM3 and COM4). Before installing the **MDPlot** software you should determine which serial ports on your machine are already in use and which are unused and available for the Digitizer. Usually you can do this by simply looking at the serial output connectors on the back of the computer to see which are connected to peripheral devices.

On some PC's the mouse is connected to a special connector that is actually one of the computer's serial ports, often COM1. To find out if this is the case, you can run a diagnostic program that displays serial port usage. If your operating system is MSDOS or Windows, run the Microsoft diagnostics tool, **msd.exe**, and note the information for the mouse and serial ports. Under Windows NT 4 or Windows 2000, open the Control Panel > Devices. If the "Sermouse" device is shown as "System" then the mouse is most likely connected to COM1. To find out, leave the Control Panel and open Admin Tools > Windows NT Diagnostics and then open Interrupts > Ports. If "PointerPort0" is assigned to I/O address 0x3F8 then the mouse uses COM1.

Once you have identified an unused serial port, connect the supplied cable between the **SERIAL OUTPUT** connector on the back of the MD3 and the computer's serial port connector. This cable will have a 9-pin male connector at one end and a 9-pin female connector at the other.

Connecting the Optical Encoders and Footswitch

Locate the X-axis encoder cable and connect it between the X-axis encoder and the MD3 at **X-AXIS INPUT**. This cable will have 9-pin male connectors at both ends and can be connected in either direction. Connect the plug from the Y-axis encoder to the **Y-AXIS INPUT** on the back of the MD3. Install the plug from the footswitch at **FOOTSW**. Make sure the MD3 power switch (front panel) is set off (O position). Connect the supplied 5 VDC adapter to the MD3 at the rear panel coaxial power jack. Plug the line cord into the 5 VDC adapter and then into a properly grounded AC outlet, preferably the same one to which you have connected your computer . Your Digitizer should now be ready for use.

Plotter or HPGL/2 Printer Connection

Digital Plotters

If using a digital plotter, connect a serial communications plotter cable (not supplied) between an unused serial port on the PC and the digital plotter. This cable will have a 9-pin connector at the computer end and a 25-pin connector at the plotter end. (If your computer is a PC or PC/XT it will have a 25-pin serial port connector, in which case see Appendix C for cable specifications for these computers).

Note: If your computer has only two serial ports, and one is used by the mouse, then you will have to do one of two things. Install a serial adapter card to expand the number of ports available, or switch between the MD3 Digitizer and the digital plotter using a serial switch box (not supplied) or by simply changing cables as needed from the Digitizer to the plotter.

Your digital plotter should be compatible with the Hewlett-Packard Graphics Language (HPGL or HPGL/2), such as used on the HP 7475A or the HP 7550A, or with the Houston Instrument DM/PL III Graphics Language. For proper serial communications refer to your plotter manual and follow the directions there to set the plotter transmission parameters (baud rate, number of data bits, number of stop bits and parity). We recommend 9600 baud, 8 data bits, 1 stop bit and no parity. These are the software defaults as specified in the resource file, *$mdprc.ini*. On the HP 7475A set the rear switches as follows for these defaults: S1 and S2 to "0", D/Y switch to D, US/MET to US, A3/A4 to A4, B4 to "1", B3 to "0", B2 to "1" and B1 to "0" (see page 3-29, HP 7475A Operation and Interconnection Manual). Other settings may be chosen if you wish, just be sure to define them in the resource file, *$mdprc.ini*.

Houston Instrument plotters will automatically match their transmission to the data being sent and no special settings are usually required on the plotter. If in doubt refer to your Houston Instrument manual.

For output to an HP plotter set the **PLOTTER_TYPE** variable in the *$mdprc.ini* file to '**HP**' or, for a Houston Instrument plotter, to '**HI**'. The software is written to use any of the available serial ports on the computer. The particular port (COM1, COM2, etc.) is specified in the file, *$mdprc.ini*.

HPGL/2 Capable Color Printers

MDPlot supports output to color printers that have HPGL/2 emulation capability. These printers include the Hewlett-Packard DeskJet 1200C/PS and 1600C/PS. For correct operation the printer must be connected to the standard parallel printer port on your computer (i.e., LPT1). The printer cable (not supplied) must support hardwire handshake. Such cables can be purchased at most computer supply stores or catalog order houses (see Appendix D: Plotter/Printer Cable Configuration, page 37).

For the HP1200C/PS or the HP1600C/PS printers, set the **PLOTTER_TYPE** variable in the resource file, *$mdprc.ini*, to '**HPDJ**'. When **PLOTTER_TYPE** is set to '**HPDJ**', the remaining serial port parameters in the Plotter/Printer section of *$mdprc.ini* are not used.

Hardware & Software Requirements

Version 4.0 of the **MDPlot** software uses one of the PC's serial ports to communicate with the MD3 Digitizer and it sends output to a Hewlett-Packard or Houston Instrument digital plotter connected via another serial port or to an HPGL/2 compatible color printer connected to the printer (parallel) port.

Screen graphics have been tested at MDC using both medium resolution (640 x 350) EGA, 16-color graphics and high resolution (640 x 480 pixels) VGA, 16-color graphics. The software automatically detects your type of graphics adapter and sets it to either EGA or VGA. A Microsoft compatible two-button mouse using Microsoft Mouse Driver Version 7.XX or higher driver software is required for operation of the program. The software must be run under DOS (MS-DOS V6.2 or higher) or as a DOS application under Windows 98, Windows NT or Windows 2000 on a PC with an 80386, 80486 or Pentium CPU.

MDPlot V4.0 requires approximately 500 Kilobytes of disk space for the executable programs, icon files and sample data files. If you install the source code an additional 500 Kilobytes of disk space will be required.

Software Installation

Your **MDPlot** software will be supplied on one 3.5" high density (1.44MB) diskette. The software will be installed in the directory *c:\mdsys4* unless you specify a different directory to the installation program, **install.exe**. Three subdirectories in *c:\mdsys4* are required: *c:\mdsys4\bin*, *c:\mdsys4\data* and *c:\mdsys4\src*. If this is the first time you have installed **MDPlot**, these subdirectories will be created for you by **install**. The executable program, **mdplot.exe**, and its configuration files will be installed in *c:\mdsys4\bin*. Sample data files are stored in *c:\mdsys4\data*, source code in *c:\mdsys4\src* and utility programs in *c:\mdsys4\bin*.

Even if this is not the first installation of **MDPlot**, we recommend you continue to use *c:\mdsys4* (or the directory you established originally) for this installation such that old software will be updated with the new release. The exceptions would be any files, such as **MDPlot** source code or the resource file, *$mdprc.ini*, that you may have modified in the meantime. In this case you should make backup copies of these files before running **install**. If **install** finds an existing copy of *$mdprc.ini*, it will back it up to the file *$mdprc.bak*

Install the software by running the program **install.exe** supplied on Distribution Disk 1. Place Distribution Disk 1 in either diskette drive A or drive B, switch to the chosen drive and then run **install**:

- In the case of drive A:
  ```
  C>  a:
  A>  install
  ```
- In the case of drive B:
  ```
  C>  b:
  B>  install
  ```
- Follow the instructions provided by the program to select the desired working directory for software installation. The default directory is *c:\mdsys4*. After the directory is specified, **install** copies the executable and sample data files to the subdirectories *bin* and *data*.
- At this point you are given the option of loading source code files. If you elect to install them they are loaded into the subdirectory *src*. If you do not want to load these files, you can always rerun **install** to load them at a later time.
- After the software has been loaded, **install** will optionally modify your *autoexec.bat* file to include the *c:\mdsys4\bin* directory in the **PATH** variable and to define two new environment variables, **MDPEXE** and **MDPDATA** that specify the directories for **MDPlot** binaries and sample data files.
- Lastly, **install** will give you the option of customizing the resource file, *$mdprc.ini* for your own hardware and plotter configuration. *$mdprc.ini* is a text file that contains **MDPlot** configuration data. **MDPlot** references it at run time to initialize a number of parameters, including your plotter type, serial port configuration, screen colors, etc. You can set certain of the parameters in *$mdprc.ini* at installation, in particular the serial port to be used by the Digitizer and the configuration of the serial port serving the digital plotter. If you change any of the default settings be sure that the appropriate hardware switches on the plotter are also set accordingly. If you want to change configuration at any time you can edit the *$mdprc.ini* file using a text editor (see Appendix B: MDPlot Resource File, page

27 for a description of *$mdprc.ini*).

When software installation is complete **mdplot.exe**, **plotfile.exe**, **ptest.exe**, *$mdprc.ini*, and the bitmap icon files (**.icn*) will have been copied to your *\mdsys4\bin* directory. Sample data files are stored in *\mdsys4\data* and, optionally, the source code in *\mdsys4\src*.

mdplot.exe (**MDPlot**) is the executable binary program for taking data, plotting on the color monitor and for output of graphics results to a digital plotter or color printer. It is the main program that you will be using and its operation is described in detail in Chapters 3 and 4.

Two utility programs are included in the *\mdsys4\bin* subdirectory: **ptest.exe** to test output to the plotter, and **plotfile.exe** to send plotter command files to a Hewlett-Packard or Houston Instrument compatible digital plotter or HPGL/2 compatible color printer. See Chapter 5 for more information on using these utilities.

A copy of this manual, in Adobe Acrobat Portable Document Format (pdf), will be found in the main directory *\mdsys4*.

Environment Variables

The two environment variables, **MDPEXE** and **MDPDATA**, define the locations of **MDPlot** executable programs and the directory for storing data files. **MDPEXE** defaults to the directory *\mdsys4\bin*. **MDPDATA** can be any valid directory name. It is initially set to *\mdsys4\data* by the **install** program; however, you may want to change the variable to point to a directory where you would normally store your own **MDPlot** data files. In MSDOS you can enter **MDPEXE** and **MDPDATA** from the keyboard using the set command:

```
C> set mdpexe=\mdsys4\bin
C> set mdpdata=\mydata\direct
```

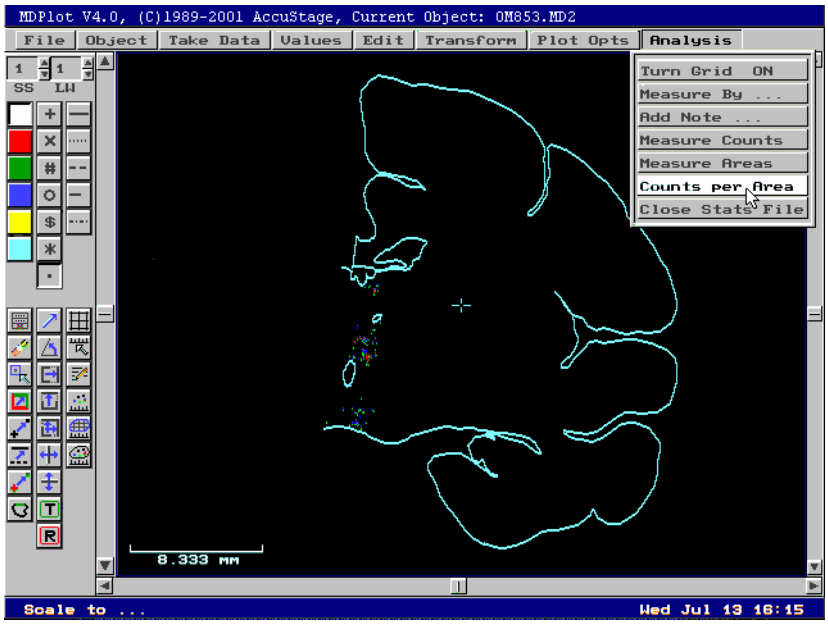
Rather than manually setting **MDPEXE** and **MDPDATA** each time you start your computer you can place the above commands in your *autoexec.bat* file.

MDPEXE is used by **MDPlot** to locate its icon files (**.icn*) and the resource file, *$mdprc.ini*. If the variable is not defined, **MDPlot** expects to find these files in the current working directory. If **MDPlot** can't locate the icon files or *$mdprc.ini* an error results.

CHAPTER 3: PROGRAM OPERATION

Program functions in **MDPlot** v4.0 are controlled through a graphical user interface (GUI) which uses the mouse as a pointing/selection device and several different types of graphical control devices including pull down menus, list boxes, dialog boxes, scroll bars, iconic push buttons, slide controls, picking cursor and bounding polygons. The control and message areas of the GUI run along the top, left side and bottom of the screen, while data are displayed in the central portion of the screen, or data window. The data window constitutes part of the GUI since graphical elements displayed here can be manipulated interactively using the mouse.

You may want to run the program while referring to the descriptions below:



Title Bar

The top horizontal panel on the screen is the program title bar and current database object listing area:

```
MDPlot V4.0e (C) 1989-2001 AccuStage, Current Object:
```

As you load each file from disk or select the data from different files from within the program, the name of the file will be displayed on this line as the Current Object. Once loaded into **MDPlot** the data from a file are referred to as a data object or simply object.

Menu Control Bar

Below the title bar is the menu control bar. Placing the mouse cursor on one of the menu buttons in this bar and then pressing the left mouse button activates a pull down menu of functions. To select from the pull down menu use one of two methods: (a) Continue to press

and hold the left mouse button while scrolling the list of functions, releasing the mouse button over the item of choice. (b) Click (rapid press and release) on the menu control bar button, select the desired item from the menu and then, with the cursor over the item of choice, click the left mouse button again.

The menu control bar lists functions by logical category:

[File][Object][Take Data][Values][Edit][Transform][Plot Opts][Analysis]

Descriptions of the functions within each category are given in Chapter 4: Program Description, page 12.

Push Button Control Area

Controls for Plot Parameters

Along the left side of the screen are two functional groupings of iconic push buttons. The group on the top left control plot parameters and the second group on the bottom left control editing and analysis functions. Within each group, buttons for any given function are arranged in a vertical column. As depicted in Fig. 3-1, the buttons for selecting plot parameters are arranged in three columns for pen colors, point symbol types and line types. The six push buttons in the first column, from top to bottom, select the symbol colors white (black on plotter), red, green, yellow, blue and magenta. The middle column controls selection of seven different point symbol types, and the third column provides selection of five different line symbol types.

At the top are controls for selecting symbol size and line width. Symbol sizes range from 1 (smallest) to 10 (largest), while line widths range from 1 to 5 with 1 being the finest width. Symbol size and line width will show on finished output to a digital plotter or printer but not on the computer screen.

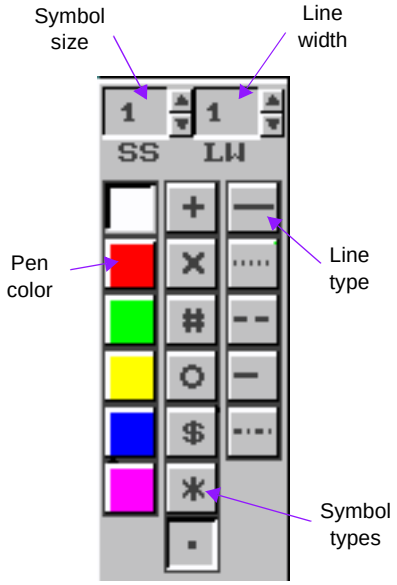


Figure 3-1. Plot symbol and line push button controls.

Note: Line width is supported only on HPGL/2 compatible printers.

Color, point symbol and line symbol buttons are radio buttons, that is, a button will remain on until another push button in the same logical group is selected. The point symbol and line symbol push buttons are also ganged across their respective groups to turn off symbols should lines be selected and vice versa.

Controls for Editing and Analysis Functions

Figure 3-2. Push buttons for editing and analysis functions.

At the bottom of the icon area on the left side of the screen are three columns of push buttons for control of the editing, geometric transformation and measurement and analysis functions (Fig. 3-2).

Editing, geometric transformation and analysis icon buttons are momentary push buttons that are turned off under program control once the function is complete.

Message Area

A single line of text space is provided at the bottom of the GUI for messages and the current date/time. Command prompts are displayed here as well as error messages. Command prompts will be preceded by a beep from the computer if the **BELL** parameter in *$mdprc.ini* is set to **ON**; otherwise, command prompts are issued silently. Error messages are always announced with a beep from the computer.

```
Scale to ...                          Wed Jul 13
```

Figure 3-3. Example of message area string.

Data Window

New data from the MD3 or data loaded from files are drawn in the data window. If the data are new from the MD3 the default scale factor (**SCALE_FACTOR**), as read from the resource file *$mdprc.ini*, will be used. If the data come instead from a disk file then they will be drawn using the scale factor stored in the file. Any new data added to an existing database, either from another file or active MD3 data, will be drawn at the existing scale factor. You control the size of a drawing in the current plotter chart frame by setting the scale factor (see 'Values Menu' on page 15).

Along the left, bottom and right sides of the data window are three slide controls that are used to pan and zoom the viewpoint (Fig. 3-4). The left control shifts your view of the data up and down, or in the Y-direction, and the bottom control shifts your view horizontally in the X-direction. The control on the right side of the window zooms your view up or down. These controls are used to position the data on the screen for convenience of viewing, to check the size of a plot before sending it to the plotter and to assist in visualizing data during editing operations. To operate the slide controls place the cursor over the slider button, press and hold the left mouse button and reposition the slider. If the slider reaches the stop at either end of the control, clicking on the button at that end allows continued movement in the same direction. This resets the slider button to the other end of the control from where you can continue movement.

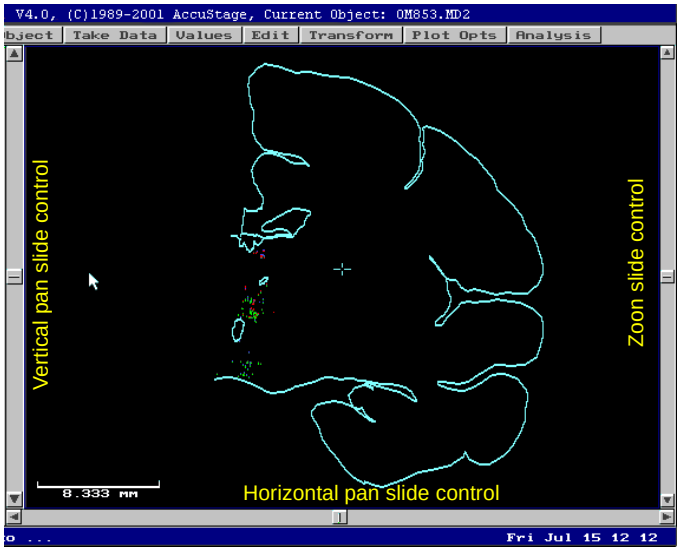


Figure 3-4. Data display window showing locations of pan and zoom slide controls.

Note: The pan and zoom controls do not affect the coordinate data in the database, but rather how they are displayed on the screen. Geometric transformations of the database are performed separately using functions such as translate, rotate, scale and flip.

The data window becomes an interactive interface during editing or analysis operations. For example, when translating data or when changing a data attribute such as symbol type, line type or color, the mouse is used to indicate the desired translation or to pick data elements directly on the screen. In all cases use the left mouse button for carrying out editing operations unless the function specifically directs you to use the right mouse button (for example, when closing a polygon).

Plot Symbol Shortcut Keys

Frequently used point symbol/color and line symbol/color combinations can be assigned to the computer keyboard function keys **F1** to **F12**, and also to the MD3 keypad keys **BEGIN**, **END**, **ENTER**, **DOT** (.) and the number keys **0** through **9**. Shortcut key values are defined in the resource file *$mdprc.ini*. To change a key value, use an editor such as Windows Notepad to edit the entry in the file.

A single press of a shortcut function key or keypad key will perform the equivalent of selecting a symbol icon/color icon pair using the mouse. This feature is most useful when taking data to change between frequently used symbols. See Appendix B: MDPlot Resource File, page 27, for the format of a key definition.

Right Mouse Button and ESC Key

In most cases the right mouse button or the **ESC** key can be used to interrupt or cancel an operation. For example, to interrupt data acquisition from the MD3 you can either select the Stop Acquisition menu item from the Take Data menu or simply click on the right mouse button. However, clicking the right mouse button after an editing operation causes a repeat of the last editing command. In this instance there will be a prompt message displayed indicating use of the right mouse button.

Entering Text in List Boxes

To enter text in list and dialog widgets you can simply click on one of the names displayed in the scrollable list area, type text directly into the text string areas of the widget, or modify text already in the string areas.

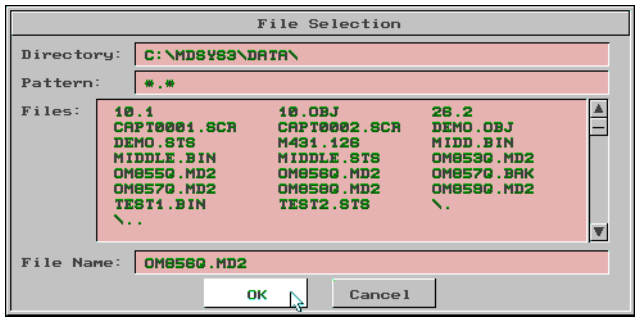


Figure 3-5. File and directory chooser.

To enter or modify text, place the mouse cursor within the text string area and click on the left mouse button. Any text already in the string will be highlighted in green. A second click on a specific location in the string will highlight a single character in green. Pressing and holding the left mouse button down while scrolling over several characters will highlight the scrolled characters. You can proceed to enter text at the beginning of the green area, or you can delete characters highlighted in green by pressing the Del key on the keyboard. The backspace, left arrow and right arrow keys can also be used to position the text cursor. When done typing text, press the Enter key on the keyboard or click the right mouse button. Finally, to accept or cancel your entry, click on either the Okay or Cancel buttons in the widget.

The Resource File, *$mdprc.ini*

Configuration and setup parameters for the **MDPlot** software are stored in a text file, called the MDPlot resource file, which resides in the *\mdsys4\bin* directory under the name, $mdprc.ini. Parameters are provided in the *$mdprc.ini* file for defining not only hardware settings for ports and plotter types, but also for defining path names for data storage, setting screen colors, default plotter chart size, plot origin and scale factor. "Speed key" definitions are also provided for assigning frequently used plotting symbol/color pairs to function keys **F1** through **F12** and the MD3 membrane keypad keys. See Appendix B: MDPlot Resource File, page 27, for a listing of a typical *$mdprc.ini* file and a description of the format of entries.

You can edit the *$mdprc.ini* file using a text editor to configure **MDPlot** for your own hardware, screen color and plot preferences. The file should be saved out of the text editor as a plain-text file, i.e., devoid of formatting characters and commands. Be sure to store the file back in the *\mdsys4\bin* directory when done.

MDPlot expects to find the *$mdprc.ini* file in the *\mdsys4\bin* directory, or in the directory specified by the **MDPEXE** environment variable. The program reads *$mdprc.ini* early in its initialization phase and sets internal program variables to values found in the file. If a variable is missing or is set to an incorrect value, **MDPlot** uses the default value built into the program for that variable. The *$mdprc.ini* file supplied with this distribution is a complete list of the variables which can be set via this mechanism.

CHAPTER 4: PROGRAM DESCRIPTION

MDPlot has been designed as an object oriented graphics editor. The object orientation takes the form of an internal database design that considers each disk file and each active MD3 data acquisition session to be a separate data object made up of graphical elements. These graphical elements can be either points or contours, and each element can have its own attributes that include symbol type and color.

A data object is created within **MDPlot** each time you load a new file or when you take data from the MD3. The program can handle up to twenty separate data objects at a time depending on available memory and the size of each individual object. You can also merge a disk file or a new data acquisition session with a previously existing data object. In this case, the merged data then become one object for purposes of editing, analysis and subsequent disk storage.

At the data object level, you have control over one of two modes of editing and analysis: (1) Apply operations to one object in particular, or (2) Apply them to all objects currently in memory. The Object menu described on p. 14 controls level modes.

When performing editing and analysis operations on data objects, different groupings of elements within an object can be selected and the edit or analysis function applied to just those elements. Both the Edit menu and Analysis menu have a function for setting the means of picking individual elements using the mouse. For example, you can pick single elements for editing or pick more than one element for analysis using rubberband boxes or polygons. The Edit menu also includes a function, Selected Elements, for selecting generic elements by symbol and/or line type.

Functions in **MDPlot** are accessed from pull down menus or from push button icons. Some functions are accessed only from menus and others only from icons. For example, selection of point or line symbol colors and types is by icon only, and file I/O is managed only from the File menu. However, many functions have both icon and menu equivalents; for instance, most of the editing, transformation and analysis functions. The menus are organized by categories, including file I/O operations, data acquisition functions, object selection, etc. A brief description of the functions in each menu follows. Where the menu functions have iconic push button equivalents these are described in the section immediately after the menu description.

File Menu

Open and Merge read **MDPlot** data files into memory and create the internal data

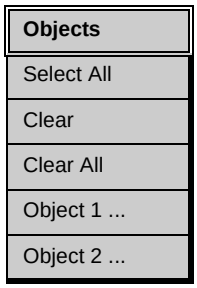
File	
Open	Open and read an existing disk file.
Merge	Merge a disk file with current object in memory.
Save Obj	Save object to disk in 'object' format.
Save Obj As ...	Save in 'object' format under new name.
Save Obj All	Save all data objects in 'object' format.
Save Bin	Save object in binary file format.
Save Bin As ...	Save in binary format under a new name.
Save Bin All	Save all data objects in binary format.
Close	Close (write and clear) current object.
Close All	Close all data objects.
Exit	Exit the **MDPlot** program.

objects. Open creates one object per data file, and adds the object to the list of objects currently in memory (see the description of the Object menu on p. 14). Merge, on the other hand, combines the file contents with the current object, as defined by the Object menu, to make one larger object. Save ... functions write the data to disk without altering memory. Save Obj, Save Obj All, Save Bin, and Save Bin All save data using the object's current file name. Any existing disk file of the same name is renamed with a *.bak* extension before the new file is written. Save Obj As ... and Save Bin As ... save the file under a new name which you enter from the directory/file listing widget. The Close function writes the current data object to disk under its current name and in its original format ('object' or binary), and then releases memory used by that object. Any preexisting file is renamed with a *.bak* extension. Exit will prompt before exiting if data objects have been modified.

MDPlot recognizes two different types of file formats, object and binary. Both types contain the exact same information, object format being a text-based database description, whereas binary format is a compact numeric database description. **MDPlot** v4.0 is backward compatible with all previous versions of the software and will read binary and object format files written by these previous versions. See Appendix C: Data File Format, page 33, for a description of **MDPlot** file formats. There are no iconic push button equivalents for file I/O functions.

Object Menu

The Object menu is used to select objects for editing, geometric transformation and measurement analysis. A menu item is created for each file that is loaded into **MDPlot** using the File Open function. To select just one data object for editing or analysis, click on the individual Object 1..., etc., menu item for the object. Selected objects will be highlighted in the



All data objects are to be included in edits and analysis.

Delete the current data object from memory.

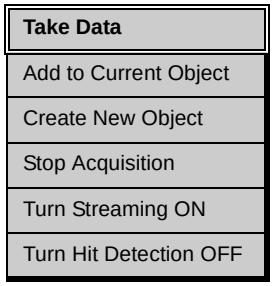
Delete all data objects from memory.

"Object 1", "Object 2", etc., give names of data objects currently in memory.

menu. If Select All is chosen, all objects will be highlighted and editing, geometric transformation and analysis functions will be applied to all the objects currently in memory. The Clear and Clear All functions delete all data for one object or for all objects, respectively. Data are cleared from memory without affecting disk files.

Take Data Menu

The Take Data menu controls acquisition of data from the MD3 Digitizer. To enter data from the MD3 and create a new data object use the Create New Object function. This function will first prompt you to locate an origin (x=0, y=0) for your specimen. The MD3 Digitizer can be used to set the origin to a point you select from your specimen, or it can set the origin to fiduciary marks internal to the optical encoders.



Append Digitizer data to object in memory.

Make a new database object for input of Digitizer data.

Stop data input from the Digitizer.

Enable/disable data streaming mode. The stream increment is set in the Values menu.

Disable/enable detection of overlapping points during data acquisition.

- To set the origin to a reference on your specimen, place that point under the crosshair in the microscope eyepiece and then press the **ORIGIN** key on the Digitizer membrane keypad.
- Use the **ORIGIN ON MARK** function of the MD3 to set the origin to the optical encoder's internal reference points. These are special marks on the glass scales in the encoders which are located approximately midway in the encoder's movement. They allow for setting the stage to an absolute position that is repeatable to within the encoder resolution (2 microns). After pressing the **ORIGIN ON MARK** key, move the stage past center and then press the MD3 **BEGIN** key. Slowly move the stage toward the mid position while observing the display. As each encoder is centered the display will read "X axis reset" and "Y axis

reset". When using this feature, you should approach center consistently from one direction since there is a finite width of the encoder reference pulse equal to the encoder resolution (2 microns).

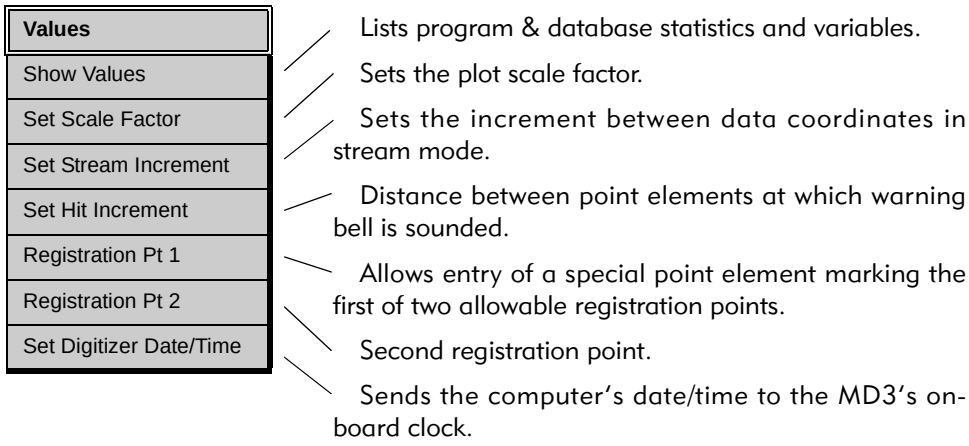
As you take data, use the icons at the top of the icon area to select colors and symbol types (page 8), or use the keyboard function or MD3 keypad keys (page 10, cf. Appendix B: MDPlot Resource File, page 27). If you change from point symbols to lines the program will automatically revert the data acquisition mode from point to line mode. For example, if you're marking points using the + symbol and you want to draw a contour using a solid line in the color cyan, simply click on the solid line symbol and then on the cyan color push button. Now, as you continue to take data you will be drawing a line element. Symbol size and line width should be set before selecting the symbol or line type. Symbol sizes and line widths are used on the finished digital plotter or color printer output. On the screen, however, symbols and line widths are drawn at the smallest size of 1 unit.

Streaming refers to the automatic entry of data whenever the stage has been moved a predetermined amount. If streaming is set ON, the data acquisition function in **MDPlot** will store a coordinate pair, or vertex, every time the stage has moved a given amount from its previous position. The amount of movement between vertices is entered using the Set Stream Incr function in the Values menu. Streaming is used most often to draw boundaries without having to push the foot switch or **SAMPLE** key on the Digitizer.

Hit detection, when enabled, warns you with a ring of the computer bell whenever a point that you have just entered falls within a predetermined distance (the hit increment) of another point element in the object database. If you want to delete the new point element, press either the DEL key on the computer keyboard or press the right mouse button before taking any further data. The hit increment is entered using the Values menu and the Set Hit Increment function.

Values Menu

Use Set Scale Factor to enter the scale factor for plotting. Units are in millimeters of plotting distance per millimeter of microscope stage movement. Positioning of the data on the plotter paper can be judged from the screen view which shows a plotter paper-sized rectangular frame drawn around the data (you may have to zoom out on the data window to see the frame). The frame size and orientation correspond to the chart size that is entered in the Set Chart Size option of the Plot Options menu (page 20). Set Digitizer Date/Time down-



loads your computer's current clock information to the microcontroller in the MD3 where it

updates the onboard, battery-powered clock. Registration Pt 1 and Registration Pt 2 let you enter special fiduciary marks that can subsequently be used, for example, to register adjacent sections (see p.).

Edit Menu

Edit	
Undo Last	Undo the last edit or geometric operation.
Erase	Erase points and/or contours.
Protocol	Enter a line of text that describes the object.
Selected Elements	Edit selected types of elements in an object.
Select By	Set the mode for choosing elements to be edited.
Change Color	Change color of selected elements.
Change Symbol Type	Change symbol type of selected point elements.
Change Line Type	Change line type of contour elements.
Change Symbol/Color	Change symbol to new symbol in new color.
Close Contour(s)	Connect first and last vertices of selected contour(s).

Except for the Protocol function, all of the editing functions operate on either the current object or on all objects as determined by the settings chosen in the Object menu (page 14).

You can display and edit selected sets of elements in an object using the Selected Elements function. The function first determines the types of symbols and lines in an object and then pops up a button box that contains a listing and a button for each type of element. To chose certain types of symbols or lines for editing simply click on the button entry for that element. In the example shown in the adjoining figure, the object OM855G.MD2 contains four different types of symbols, which are white, red, green and blue dots, and one type of line element represented by solid white lines. The elements represented by green dot symbols and solid white lines have been selected for editing and, in addition, the user has elected to make the origin mark visible. The Selected Elements function can also be used to display and plot subsets of elements in an object. When you have two or more objects in memory, the function lets you choose sets of elements separately for each object.

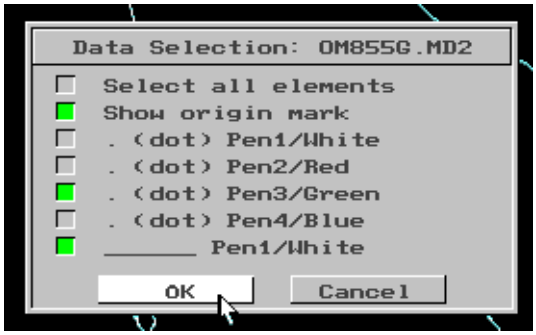


Figure 4-1. Checkbox used to select elements for editing and analysis.

The Select By ... function controls how you pick elements within data objects for editing. It presents you with a dialog box containing the selections:

Each	Modify only individually picked elements.
Box	Edit elements within a rubberband box.
Polygon	Edit elements within a bounding polygon.
All	Edit all elements in the current object(s).

Picking mode Each expects you to pick an individual point or contour element for editing using the mouse cursor. For example, with Select By ... in Each picking mode, choose an editing function such as Change Color. Place the tip of the cursor on the point or contour and click the left mouse button. The individual point or contour will then be selected for color change.

Box picking mode expects you to scroll out a rubberband box to pick elements for editing. Position the mouse cursor at one corner of the desired box location. Next press and hold the left mouse button and then drag the opposite corner of the box to the desired location. Let up on the mouse button to perform the edit.

Polygon picking mode lets you draw a rubberband polygon around the desired elements by dropping vertices of the polygon with clicks on the left mouse button. To close the polygon click on the right mouse button.

All picking mode picks all elements in the current data object(s) for editing.

Change Color, Change Symbol Type, Change Line Type and Change Symbol/Color Pair change attributes of those elements in the current data object(s) as picked using the existing Select By ... picking mode and selected by the Selected Elements function. Each function will ask you to choose the attribute to be changed from among the icons in the upper icon area of the screen (page 8). Next they ask for the new attribute to be applied.

Close Contour(s) connects the first and last vertices of a line element to form a closed contour. Closed contours are required for the analysis functions of counts within a contour, contour areas, and counts per unit area of contour (see Analysis menu, page 21).

The Undo function undoes the effects of the last editing or geometric transformation operation, or, if Take Data is in effect, removes the most recently entered Digitizer data. Undo works in two modes, one while you have the program in Take Data mode and the other when editing or geometrically transforming the database. To undo data entries you should already be in, or should first place the program in, data entry mode (see Take Data menu, page 14). To undo editing operations you should be out of data acquisition mode. Each editing or geometric transformation operation is "memorized" in an internal "undo buffer" in the **MDPlot** program. Successive Undo's of editing or transformation operations will continue to back up the chain of data editing operations, returning the state of the program to an earlier one. The size of the undo buffer, i.e., the number of operations that can be memorized, is set by the **UNDO_BUFFER** parameter in *$mdprc.ini*. In some cases, it will not be possible to undo the effects of geometric operations when you have performed an intermediate Save Transformation.

Edit Icons

The column of icon push buttons on the left side of the lower icon area offer quick, "shorthand" activation of a number of the same editing functions as just described in the previous section on the Edit menu.

UNDO

ERASE

SELECT BY

CHANGE COLOR

CHANGE SYMBOL TYPE

CHANGE LINE TYPE

CHANGE SYMBOL/COLOR

CLOSE CONTOUR

Geometric Transformations Menu

Transform	
Translate	Interactive translation of data object(s).
Rotate	Interactive rotation.
Scale X	Interactive scaling of X coordinates only.
Scale Y	Interactive scaling of Y coordinates only.
Scale All	Scaling of both X and Y coordinates.
Flip X	Multiplies X coordinates by -1 (“mirror X“).
Flip Y	Multiplies Y coordinates by -1 (“mirror Y“).
Register	Registers objects using registration points 1 and 2.
Save Trans	Applies current transformation matrix to data coordinates.
Reset	Resets the transformation matrix to an identity matrix.

The geometric transformations modify a 2-D geometry matrix that then multiplies the coordinate data of the current object or objects before they are displayed on the screen. Actual data coordinates are not altered unless a Save Transformation is performed (or icon-button "T", page 20, is pressed), in which case the geometry matrix is applied to the database proper. Reset (or icon button "R", page 20) will replace the geometry matrix with an identity matrix and thereby cancel any current geometric transformation. Transformations affect the current object or all objects as determined by settings in the Object menu (page 14). The Select By ... function is not used with geometric transformations.

Geometric transformations are separate from the viewpoint transformations which are controlled by the pan and zoom sliders around the data window. Viewpoint transformations do not affect the database per se, only how the data are drawn on the screen.

Geometric operations can be used in two ways: (1) To reposition the data for a better view of elements; for example, during the editing operations of change in color or symbol type or for placement on a plotter output chart. (2) To modify the geometry of the data proper; for example, in registration of one section with another. In the first instance, you would want to Reset the transformation matrix (icon "R") after performing your editing operations, whereas in the second instance you would want to transform coordinates (icon "T").

To perform a geometric operation, use the mouse cursor and left mouse button to select points in the data window that define beginning and ending coordinates for the particular operation. For example, to perform a rotation first pick a rotation center point and then pick two points that define the angle of rotation. Similarly, scaling requires that you first pick a point representing the origin for the scale operation and then two additional points defining the initial and final coordinates of the scale operation. Flipping simply interchanges x or y coordinate values about the origin of the coordinate system and requires no interactive input.

Registration will carry out a combined set of translation, rotation and scaling transformations to place one object in geometric registration with another, and is designed to be used on serial section data in which each section is a separate data object. The function makes use of the two registration points that you can record for an object (Registration Pt 1 & 2, Values menu, page 15). It assumes that, (1) Each object has registration points 1 and 2 already recorded, (2) Registration points 1 and 2 in each object represent common respective landmarks, and, (3) The object to be registered immediately follows the reference object in the Object menu list. To register a set of serial section objects using registration points, first load the files into **MDPlot** in serial section order. (The number of files that can be loaded at once is 20 or less depending on available memory). Begin registration by selecting the second object in the Object menu list. Next, select the Register function from the Transform menu. Then click on the Save Transformation menu function (or "T" icon) to actually transform the database for the object. Repeat on the next object in the list.

Geometric Transformations Icons

TRANSLATE

ROTATE

SCALE X

SCALE Y

SCALE ALL

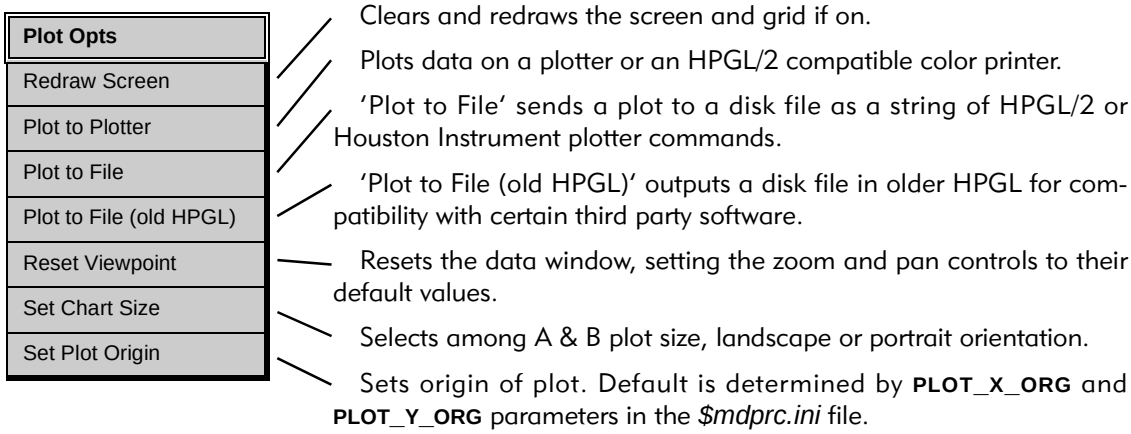
FLIP X

FLIP Y

TRANSFORM

RESET

Plot Options Menu



Plot to Plotter sends either the current object or all objects, depending on the state of the Object menu, to the digital plotter or HPGL/2 compatible color printer.

Plot to File redirects plotter commands to a disk file. The file will contain the plotter language commands in either HPGL/2 or Houston Instrument format depending on your choice of plotter type (see Appendix B: MDPlot Resource File, page 27). Plots can be stored as single files or appended to an existing plotter output file. The latter feature can be used to superimpose several plots on one chart. Plotter files can be sent to the plotter using the utility **plotfile.exe** (page 24).

Reset Viewpoint returns the data window pan and zoom viewing slider controls to their default positions. The viewing controls are also automatically reset whenever you change the scale factor or the plotter chart size.

Set Chart Size selects one of four paper sizes for plotting:

1 = Size A (US, 8-1/2 x 11 in.) or A4 (MET, 210 x 297 mm) landscape orientation.

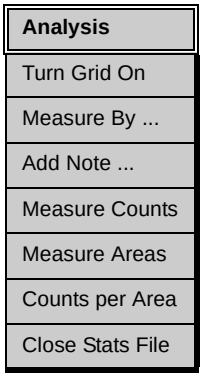
2 = Size B (US, 11 x 17 in.) or A3 (MET, 297 x 420 mm) landscape orientation.

3 = Size A (US, 8-1/2 x 11 in.) or A4 (MET, 210 x 297 mm) portrait orientation.

4 = Size B (US, 11 x 17 in.) or A3 (MET, 297 x 420 mm) portrait orientation.

Analysis Menu

The analysis functions make measurements of numbers of point elements, numbers of point elements per unit area, and areas of closed contours and boundaries.



Toggles a millimeter grid screen overlay.

Controls selection of elements for analysis.

Add a comment line to the statistics file.

Counts numbers of point elements.

Calculates areas of closed contours and boundaries.

Calculates numbers of point elements per unit area.

Closes the current statistics file.

Selection of elements is controlled by the Measure By ... function. Clicking the Measure By... menu item pops up a button box containing the choices:

Contour	Use an existing closed contour for a boundary.
Box	Use a rubberband box as boundary.
Polygon	Use a rubberband polygon as boundary.
All	Make measurements on all elements in current data object(s).

Measure By ... Contour assumes the use of a bounding contour for the ROI. The bounding contour must be closed and be an element in the current group of editable database objects. Modes All, Box & Polygon select symbols from all database objects, from a bounding rectangle or from a bounding polygon.

The Measure Counts and Counts per Area functions tabulate counts of point symbols by type and color within the region-of-interest (ROI) defined by the current Measure By ... setting.

Counts per Area calculates the numbers of point elements per unit area of the associated ROI. Units are numbers of points/mm^2. The Measure By ... setting must be one of Contour, Box or Polygon.

The Measure Areas function calculates areas of closed boundaries in millimeters squared. Measure By ... mode Contour calculates the area of a single closed contour. The contour must be an element in the current group of editable database object(s). Mode All produces areas of all contours in the current object(s). Box and Polygon modes calculate the area of a box or a polygon ROI.

Comments can be added to the statistics file using the Add Note... function. The function presents you with a combination text string and list box from which you can enter a one-line comment or select a comment from a list of your previously entered comments.

Tabulated results and comments are stored in a statistics file. This file is a text file that can be read using a word processor or entered into a spreadsheet for further data analysis. The first time you select one of the analysis functions you will be prompted to enter a file name for results. This can be an existing or new file. If an existing one, the file can be cleared first or results can be appended at the end of the file. When the statistics file is first created a brief header is placed at the beginning of the file giving the name of the file and a date/time stamp.

```
#
#       C:\MDSYS4\DATA\TEST2.STS: Fri Jul 08 08:08:23 1994
#
```

The header and any comments you might later enter are set off by a # character. For example, a comment line would appear as

```
#       This is a comment ...
```

Counts of point elements and counts per unit area are listed in the file as

```
COUNT,S1,P1,nnnnn,S1,P2,nnnnn, ...
```

or,

```
DENSITY,S1,P1,nnnnn,S1,P2,nnnnn, ...
```

COUNT and DENSITY are keywords that identify the entry as a straight count or as a count per unit area record. The label S1 through S7 indicates point symbol types 1 to 7, label P1 through P6 indicates pen color 1 to 6, and the field 'nnnnn' is the associated count. For example, the string

```
COUNT,S1,P3,45,S7,P1,98
```

indicates 45 symbols of type 1 in pen color 3, and 98 symbols of type 7 in pen color 1, whereas, the string

```
DENSITY,S2,P4,0.3489
```

indicates a density of 0.3489 counts per mm^2 of symbol type 2 and pen color 4.

Area measurement entries have the format

```
CONTOUR_AREA,L1,P1,nnnnn,L1,P2,nnnnn, ...
BOX_AREA,nnnnn
POLYGON_AREA,nnnnn
```

CONTOUR_AREA is a keyword that identifies the entry as a contour area record, BOX_AREA identifies a box ROI area, and POLYGON_AREA a polygon ROI area. In the case of CONTOUR_AREA, the label L1 through L5 indicates line types 1 to 5, label P1 through P6 indicates pen color 1 to 6, and the field 'nnnnn' is the associated area. For example, the string

```
CONTOUR_AREA,L1,P3,11.235,L2,P1,33.24E-04
```

describes results for two contours, one of line type 1, pen color 3, having an area of 11.235 mm^2, and a second for a contour of line type 2, pen color 1 having an area of 0.003324 mm^2. The string

```
POLYGON_AREA,75.43E+02
```

describes a polygon ROI of area 7543.0 mm^2.

Analysis Function Icons

GRID ON/OFF

MEASURE BY

ADD NOTE ...

COUNTS

AREAS

COUNTS PER AREA

CHAPTER 5: MDPLOT UTILITIES

Plotter Test Program - ptest.

ptest.exe is a diagnostic routine for output of a test pattern to a digital plotter connected to a serial port. The program initially lists the port number being used (COM1, COM2, ...), number of data bits, number of stop bits, parity, baud rate, and the plotter type (Hewlett Packard or Houston Instrument) as given in the resource file, *$mdprc.ini*. Next the program draws a figure consisting of 360 line segments centered at **X = PLOT_ORG_X**, **Y = PLOT_ORG_Y**. A message string is then drawn below the figure.

Approximately 5,000 ASCII characters are sent to the plotter during the course of program execution. This results in about 15 handshake operations for a plotter buffer size of 1024 as on the HP7475. Should the plotter not be connected properly or is unresponsive, error messages are printed on the monitor. Error logging is also enabled such that the routine queries the plotter with every HPGL/2 command sent using the OE command. If errors are reported back by the plotter, the offending command, HP error code, the date and time of the error are stored in a disk file under the name, *errlog*. This file can be read with the DOS type command or the Windows Notepad program.

The most common problem encountered with the plotter is improper cable wiring. See Appendix D: Plotter/Printer Cable Configuration, page 37, for configuration of the serial cable.

In cases when the plotter 'hangs-up' part way through a plot, and cabling and plotter settings are otherwise correct, it often helps to lower the transmission speed from 9600 baud to 4800 or even lower. Edit the *$mdprc.ini* file to change the baud rate. Then be sure to change switch settings on the plotter, if required, to select the corresponding baud rate (Houston Instrument plotters will automatically adjust their baud rate to the received signal).

Off-line Plotting of Files - plotfile.

Files in either HPGL/2 or Houston Instrument format that have been saved from **MDPlot** (Plot to File, Plot Options menu, page 20) can be output to a Hewlett-Packard or Houston Instrument compatible plotter or to an HPGL/2 compatible color printer using the plotfile utility. MSDOS command line format is either,

```
C> plotfile
```

in which case the program prompts you for file names, or,

```
C> plotfile [-p] file1 [file2 file3  ...]
```

where *file1*, *file2*, ... are **MDPlot** plotter output file names. File names may include directory components and also wild card constructs. Option p, if given, indicates no pause between plots. Default is to pause before output to the plotter to allow paper change.

For example, the following MSDOS command would be a valid command to plotfile:

```
C> plotfile file1.plt \mdsys4\data\*.plt
```

It would plot the file *file1.plt* in the current directory and all files in the directory *\mdsys4\data* with file name extensions *.plt*. The program would pause before each file is plotted.

APPENDIX A: MDPLOT SOURCE CODE

The **MDPlot** software consists of the main program module *mdplot.c* and a number of separate C language functions. Descriptions of *mdplot.c* and the functions it calls are given in the source code listings. The following is a brief synopsis of the major functions in **MDPlot**:

analysis.c	-	Count & area measurement.
beep.c	-	Beeps the console bell.
device.c	-	Low level digitizer, mouse & plotter routines.
dialog.c	-	Text entry/radio button widget.
dirlist.c	-	Directory/file widget.
graphics.c	-	Screen graphics functions.
gtext.c	-	Graphics mode text string I/O.
mdglbls.c	-	Global variables for all functions.
mkdirlst.c	-	Makes directory listing for dirlist().
notelist.c	-	Adds comments to the statsitics file.
objedit.c	-	Performs edits on database elements.
objfile.c	-	File I/O functions.
objmem.c	-	Object memory management functions.
objselct.c	-	Object selection.
objvalue.c	-	Lists database values.
plotdata.c	-	Plots object data. Initializes matrices.
plotsel.c	-	Selects elements for editing and plotting.
resource.c	-	Reads *$mdprc.ini* & initializes variables.
sleep.c	-	Timed pause.
speedkey.c	-	Speed key handler.
takedata.c	-	Digitizer input routine.
textstr.c	-	Title & message string output.
undo.c	-	Undo functions.
utils.c	-	Error listing utilities.

Additional graphical widgets and geometric functions are provided as object modules:

icons.obj	-	Icon widgets.
pupmenus.obj	-	Popup menu widgets.
widgets.obj	-	Text, list and button widgets.
transfrm.obj	-	Geometry functions and matrix math.

Data are taken from the MD3 by **device** and **takedata** and are displayed on the computer monitor by **plotdata**.

Screen graphics are managed by **plotdata** through calls to a number of functions in **graphics**. Data are output to the plotter or printer by functions in **device**.

The GUI makes use of a number of functions including low level widgets in **widgets,** icons, and mouse routines in **device**, and higher level widgets such as **pupmenus**, **dialog** and **dirlist**.

File input and output is handled by **objfile** and database memory allocation and deallocation are managed by **objmem**.

The geometry functions in **transfrm** are used extensively by the screen and plotter graphics routines as well as by the editing and measurement functions in **objedit** and **analysis**.

Source code modules for **icons**, **pupmenus**, **textwidg** and **transfrm** are not included in the software; however, these modules are supplied in object form. Examples of their use can be seen in the functions that call them, such as **dirlist**, **objedit** and **mdplot**, and templates for their controlling structures are found in **gl.h**.

mdplot is primarily a "driver" program that performs initialization and then handles events from the GUI, dispatching these to specific event handlers such as **objedit**, **analysis**, **takedata**, etc. Global variables are used rather extensively and are found in the file *mdglbls.c* with external definitions in *mdglbls.h*.

Should you care to modify the source code for any reason you must have a Microsoft C compiler and the serial communications library, CommLib v5.2, from Greenleaf Software, (Greenleaf Software, Inc., 16479 Dallas Parkway, Suite 570, Dallas TX 75248, (214)248-2561). Version 4.0 of **MDPlot** has been compiled using Ver. 6.0 of the Microsoft Visual C/C++ compiler. The **makefile** used in that compilation is included with the source code in the file *mdplot.mak*. All modules should be compiled under the large memory model of the compiler (/AL switch) and a stack size of 8192 bytes. Set all optimizations off (compiler switch /Od) for proper compilation of *device.c* and *graphics.c*. AccuStage will not be responsible for maintaining software that has been modified by the customer.

APPENDIX B: MDPLOT RESOURCE FILE

```
#-------------------------------------------------------------------------
#
# $MDPRC.INI - MDPlot Resource File for Model MD3 Digitizer
#
# MDPlot Version 4.0
# March 2001
# Copyright (C) 1999-2001 by AccuStage
# All Rights Reserved
#
# The Resource File is used to set run-time variables in the MDPlot
# program. Variables whose values can be set include MD3 setup and
# communications parameters, plotter type and communications
# parameters, screen colors, and directory path names for MDPlot
# executable and data files.
#
# Each entry in the file has the form:
#
#         KEYWORD                 VALUE
#
# where KEYWORD is the name of the variable and VALUE its value. A
# sharp sign (#) anywhere on a line introduces a comment (such as
# this). All characters from # to the end of the line are ignored
# by MDPlot when it reads $MDPRC.INI. Blank lines can be used for
# readability and are similarly ignored by MDPlot.
#
# The KEYWORD uniquely identifies its associated VALUE to the
# MDPlot program and should not be modified. However, the VALUE of
# a variable can be edited and set to an appropriate string or
# numerical value depending on context. Errors in keywords or
# values are ignored.
#
# The $MDPRC.INI file should reside in the MDPlot home directory,
# which defaults to C:\MDSYS4\BIN at installation time. The MDPlot
# home directory can also be set in the environment using the DOS
# SET command:
#
#         > SET MDPEXE=C:\MDSYS4\BIN
#
# For convenience you may want to place this command in your
# AUTOEXEC.BAT file. MDPlot looks for the $MDPRC.INI file in the
# current working directory first. If it fails to find the file
# in the current working directory it next searches in the
# directory specified by the MDPEXE environment variable.
#
# NOTE: MDPlot also recognizes the aliases $MDPRC and $MDPRC.TXT
# as valid names for the resource file.
#
#-------------------------------------------------------------------------

# -------------------------------------------------------------------------
```

```
# Set the path for MDPlot executable
# ------------------------------------------------------------------------

MDPEXEc:\mdsys4\bin

# ------------------------------------------------------------------------
# Set the path for data files
# ------------------------------------------------------------------------

MDPDATA         c:\mdsys4\data

# ------------------------------------------------------------------------
# Digitizer Parameters
# ------------------------------------------------------------------------

DIGITIZER       MD3     # Model of Digitizer.
DIGITIZER_PORTCOM1# Serial port. Values are COM1, COM2, etc.
DIGITIZER_RES2.0        # Encoder resolution in microns.
X_POLARITY    1         # Polarity of X & Y axes. Can be 1 or -1.
Y_POLARITY    1
INCREMENT_VALUE100# Distance in microns between data reports when
                        # MD3 Digitizer is in Stream Mode. Valid
                        # distances are 2 to 1998.

# ------------------------------------------------------------------------
# Plotter/Printer Setup. Type, baud rate, serial port, etc.
# ------------------------------------------------------------------------

PLOTTER_TYPEHP      # This can be one of:
                        #  'HP'   - for Hewlett-Packard plotters
                        #           such as the HP7475 or HP7550,
                        #           or HPGL compatible laser printers.
                        #  'HPDJ' - for the Hewlett-Packard DeskJet 1200C
                        #               1200C/PS, 1600C or other HPGL/2
                        #           compatible color printers.
                        #  'HI'   - for Houston Instrument plotters
PLOTTER_PORTCOM1# Serial ports 1, 2, ... are COM1, COM2, ...
PLOTTER_BAUD9600  # Acceptable baud rates are 300, 600, 1200, 2400,
                        #    4800, or 9600
PLOTTER_STOPB1      # Number of stop bits - 1 or 2
PLOTTER_DATAB8      # Number of data bits - 7 or 8
PLOTTER_PARITYNONE# Parity - EVEN, ODD or NONE
PLOTTER_WAIT_TIME 10# Sets the maximum time in seconds to wait on a
                        #    handshake operation with the plotter. If
                        #    plotter fails to respond in this time, it
                        #    is taken off-line and the error message:
                        #    ** Plotter transmission error **
                        #    is displayed.

# ------------------------------------------------------------------------
# Color Palette
#
# The following define the default color palette values.              The color
# palette consists of 16 colors on the EGA or VGA graphics adapters.
```

```
# Each palette value consists of three bytes, one byte each for the
# red, green and blue color components. A single color component can
# vary from 0 to 63 (decimal), or 0 to 3F (hexadecimal).
#
# A color palette value has the format:
#
#            Blue Green Red
#             00    00    00
#
# For example, 00003F is full red with no green and no blue.
#
# You may change the color component values; however, the KEYWORD
# must not be changed. If you change the color BLACK the program
# uses your new definition for the color BLACK in all those
# instances where BLACK is listed in the Color Assignments Table
# (see below).
# ------------------------------------------------------------------------
# Color          B G R
BLACK            000000              # Color Index 0
DK_GRAY          101010              # Color Index 1
GRAY             303030              # Color Index 2
DK_BLUE          100000              # Color Index 3
MED_BLUE         200000              # Color Index 4
LT_BLUE          3F1010              # Color Index 5
DK_GREEN         001800              # Color Index 6
CYAN             3F3F20              # Color Index 7
TURQUOISE        202010              # Color Index 8
GREEN            002800              # Color Index 9
LT_GREEN         203F20              # Color Index 10
YELLOW           003F3F              # Color Index 11
MAGENTA          3F003F              # Color Index 12
RED              00003F              # Color Index 13
ROSE             2C2C39              # Color Index 14
WHITE            3F3F3F              # Color Index 15

# ------------------------------------------------------------------------
# Color Assignments
#
# Use of color in MDPlot varies by function. For example, six pen
# colors are provided which are referred to by the variable names
# PEN1, PEN2, etc. Colors of the screen plotting area and some of
# the menu and text widget colors can be specified as well.
# The following table is used to assign one of the color palette
# values to a functional color.
# ------------------------------------------------------------------------

PEN1             WHITE              # PEN1 is WHITE on the screen and BLACK
                                    # on the plotter or printer.
PEN2             RED                # PENs 2 thru 6 ...
PEN3             GREEN
PEN4             YELLOW
PEN5             LT_BLUE
PEN6                    MAGENTA
GR_BORDER               LT_BLUE     # Screen graphics border color
```

```
GR_BKGRND       BLACK      # Screen graphics background color
GR_TEXT         WHITE      # Screen graphics text color
MENU_PANEL      GRAY       # Background color
MENU_TEXT_OFF   DK_GRAY    # Text color of an inactive menu item
MENU_TEXT_ON    WHITE      # Text color of an active menu item
MENU_CURSOR     WHITE      # Active menu item button color
TXWG_BKGRND     ROSE       # Widget text field background color
TXWG_HILITE     WHITE      # Widget text field highlighting color
TXWG_SELECT     LT_GREEN   # Widget text field select color
GRID_COLOR      DK_GRAY    # Color of mm grid and plotter frame
RUBBERBAND      CYAN       # Color of rubber band lines and boxes

# ------------------------------------------------------------------------
# "Speed Key" Assignments
#
# Frequently used point and line symbol/color combinations
# can be assigned to the keyboard function keys, F1 thru F12,
# and to certain of the MD3 Digitizer keypad keys such that when
# one of these keys is pressed a point symbol or line symbol and
# its color are selected.
#
# KEYWORD can be one of the mnemonics:
#       For the computer keyboard:
#               F1, F2, ... F12.
#       For the MD3 keypad:
#               K_ENTER, K_END, K_BEGIN, K_DOT, K_CLEAR and
#               K_0 through K_9.
# where, in the case of the MD3 keypad, the mnemonics correspond
# to the ENTER, END, BEGIN, . (DOT), CLEAR, and 0 through 9 keys.
#
# The VALUE field specifies either a point symbol type and its
# color, or a line symbol and its color. Point symbols may be one
# of S1, S2, ... S7, where:
#       S1 = +
#       S2 = x
#       S3 = #
#       S4 = o
#       S5 = $
#       S6 = *
#       S7 =.
# line symbols one of L1, L2, ... L5, where:
#       L1 = _____
#       L2 = .....
#       L3 = __ __ __
#       L4 = ___  ___
#       L5 = __ . __ .
# and colors (or pens) one of P1, P2, ... P6 for the colors white
# (black on plotter), red, green, blue, yellow and cyan.
#
# The VALUE field for a point or line symbol and its color is
# formed by concatenating the codes for each. For example, point
# symbol 1, pen 1 has a VALUE of S1P1. Line type 3 in color 4
# would be L3P4. The order of the pairs is unimportant. S1P1 is
# equivalent to P1S1.
```

```
# ------------------------------------------------------------------------

# Computer keyboard 'speed' keys
F1      S7P1    # selects point symbol 7 (.) in pen color 1 (wh/blk)
F2      S7P2    # dot in red
F3      S7P3    # dot in green
F4      S7P4    # dot in blue
F5      S7P5    # dot in yellow
F6      S7P6    # dot in magenta
F7      S1P1    # plus in wh/blk
F8      S1P2    # plus in red
F9      L1P1    # F6 selects a solid line in white/black
F10     L1P2    # solid line in red
F11     L1P3    # solid line in green
F12     L1P4    # solid line in blue

# MD3 keypad 'speed' keys
K_1     S7P1    # selects point symbol 7 (.) in pen color 1 (wh/blk)
K_2     S7P2    # dot in red
K_3     S7P3    # dot in green
K_4     S7P4    # dot in blue
K_5     S7P5    # dot in yellow
K_6     S7P6    # dot in magenta
K_7     S1P1    # plus in wh/blk
K_8     S1P2    # plus in red
K_9     S1P3    # plus in green
K_0     L1P1    # selects a solid line in white/black
K_DOT L1P2      # solid line in red
K_CLEAR         L1P3    # solid line in green
# K_ENTER, K_END, and K_BEGIN unassigned

#------------------------------------------------------------------------
# Run Time Variables
#
# All variables in MDPlot have default values that are used
# when the program is first started. These remain in effect
# until you change them, typically from a menu and dialog
# box. However, since the defaults in MDPlot may not be
# convenient for your own application, provision is made in
# the $MDPRC file       for providing your own settings. For example,
# the default plotter (x, y) origin is (1310, 1050). By
# entering different values for the PLOT_X_ORG and PLOT_Y_ORG
# you can set the plot center to the desired coordinates.
# ------------------------------------------------------------------------

CHART_SIZE   1       # Size of plotter paper. Allowable values are:
                     # 1 = Size A (US) or A4 (MET) landscape.
                     # 2 = Size B (US) or A3 (MET) landscape.
                     # 3 = Size A (US) or A4 (MET) portrait.
                     # 4 = Size B (US) or A3 (MET) portrait.
                     # Note: If using a Hewlett-Packard plotter
                     #       be sure to set the plotter switches
                     #       accordingly.
SCALE_FACTOR10       # Scale factor of 10 (10 mm plotter = 1 mm data).
```

```
PLOT_X_ORG  1295     # Plotter X-origin in plotter units.
PLOT_Y_ORG  965      # Plotter Y-origin in plotter units.
                     # These values place the origin in the center of
                     # an 8-1/2" x 11" sheet (CHART 1). Values for
                     # the center of an 11" x 17" sheet are 2020,1295.
                     # Values assume landscape orientation and are
                     # automatically adjusted for portrait.
SUBPLOT_X_ORG1295# Origin of any subregion of a plot. Normally
SUBPLOT_Y_ORG965 # equal to PLOT_X(Y)_ORG.
SUBPLOT_SCALE10      # Scale factor of the subregion.
HIT_INCR_X     10    # X,Y size in microns of hit window. The hit
HIT_INCR_Y     10    # window is used to detect overlapping points.
                     # If a newly entered point is within HIT_INCR_X,
                     # HIT_INCR_Y of an existing point, the bell
                     # is sounded and you can delete the point using
                     # the DEL key or right mouse button. Hits do
                     # not apply to line/contour vertices. To turn
                     # hit detection off, set values to 0.

#-------------------------------------------------------------------------
# Miscellaneous Variables
#-------------------------------------------------------------------------
BELL    OFF      # ON or OFF. If ON, command prompts are
                 # accompanied by a ring of the console bell.
                 # Errors always ring the bell.
UNDO_BUFFER50    # Size of the undo buffer, i.e. the maximum number
                 # commands that can be undone.          Buffer size is
                 # limited only by the available memory after
                 # allocations for data objects, etc. Each undo
                 # buffer entry is about 44 bytes. A size of 50
                 # for this parameter is reasonable.
```

APPENDIX C: DATA FILE FORMAT

The data files stored by **MDPlot** v4.0 can be in either binary or ASCII format. A binary file contains a 256-byte header, followed by vertex, point element and contour element data. **MDPlot** reads old v2.x and v3.x binary files and converts them to the v4.x binary format when they are rewritten to disk.

Binary File Format

The 256 byte header is a C-structure having the definition:

```
struct HEADER {                 // FILE HEADER TEMPLATE
        short idcode;           // ID Code. 400 (decimal) for v4.x files
        short wdx;              // Count of data pairs written to database
                                // Used for V2 files. For v3.x and v4.x data see
                                // npoints, nvertices & ncontours below
        short lnbar;            // Length of scale bar
        short chart;            // Chart size
        short x0, y0;           // X-Y absolute origin in plotter units
        double scale;           // Plot scale factor
        double scbar;           // Scale bar value in mm
        char prot[150];         // File descriptor/protocol
        short xcntr, ycntr;     // X-Y relative plot center
        short xreg1, yreg1;     // Registration point 1 (v3.x data only)
        short xreg2, yreg2;     // Registration point 2 (v3.x data only)
        ushort nvertices;       // Number of vertex structures to follow
        ushort npoints;         // Number of point elements
        ushort ncontours;       // Number of contour elements
        float encoder_res;      // Optical encoder resolution. 2 microns for MD3
        short subx0, suby0;     // Sub-origin
        float XRegPt1, YRegPt1;         // Registration points in v4.x
        float XRegPt2, YRegPt2;
        char unused[36];        // Reserved for future versions
};
```

x0, y0, xcntr, ycntr, xreg1, yreg1, xreg2 and yreg2 are in units of Houston Instrument plotter increments (10 units = 1 millimeter of plotter movement). The type "ushort" is unsigned short.

Vertex data are stored immediately after the header as an array of C structures. Each structure has the definition:

```
struct VERTEX {
        float x;                // X-value in microns
        float y;                // Y-value in microns
};
```

The nvertices member of struct HEADER gives the number of vertices.

Immediately following the vertex data are point element data as an array of C structures. Each point element has the definition:

```
struct POINT {              // Point element */
        ushort  vertex;  // Base 0 Index to vertex */
        ushort  symbol;
        // The 4 hex digits, 0x0000, of 'symbol' are:
        //  0x   0              0              0              0
        //       |              |              |              |
        //       flags,         pen #,         mark type,     size
        // 'Flags' are used internally and will be zero in the database
        // Pen # is 1 to 6.
        // Mark type is 1 to 7, with 7 being a dot.
        // Size is 1 to 10.
};
```

The npoints member of struct HEADER gives the number of point elements.

Contours follow the point elements. Each contour entry in the file consists of a structure followed by the list of vertex indices for that contour. The contour structure has the definition:

```
struct CONTOUR {                    // Contour or line element
        ushort          nvert;      // Number of vertices in line or contour
        ushort          maxverts;   // Max size of varray
        ushort          *varray;    // Pointer to an array of vertex indices
        ushort          symbol;
        // The 4 hex digits, 0x0000,  of 'symbol' are:/
        //  0x   0              0              0              0
        //       |              |              |              |
        //       flags,         pen #,         line type,     width
        // 'Flags' are used internally and will be zero in the database.
        // Pen # is 1 to 6.
        // Line type is 1 to 5, with 1 being a solid line.
        // Width is 1 to 5 with 1 being the smallest.*/
};
```

Immediately after the structure is an array of short integers (2 bytes per value) of length CONTOUR.nvert that are indices to vertices that define the contour. The ncontours member of struct HEADER gives the number of contour elements.

Vertices have an implicit index that is the position of the vertex in the vertex list. The first vertex in the list is index 0 and the last has an index =nvertices - 1. The vertex for a point element (structure member vertex) is an index into the vertex list. Similarly, contours are described by a list of vertex indices.

Object File Format

Object format files are ASCII files having the same general organization as that of a binary file except that the header, vertex, point and contour descriptions are text based. The following is an abbreviated example of a typical object format file:

```
#
# Sat Dec 05 12:21:58 1999
# File: C:/MDSYS4/DATA/10.OBJ
#
PROTOCOL slide 10, sec 1; FB=2,HRP=3,FB cells=4,HRP cells=5
NVERTICES 550  NPOINTS 118  NCONTOURS 2
CHART_SIZE 1  PLOT_ORG_X 1300  PLOT_ORG_Y 1000
SCALE_FACTOR 10.000000
ENCODER_RES 2.00
REL_X_CNTR 0  REL_Y_CNTR 0
REG_PT1_X 0.000000  REG_PT1_Y 0.000000
REG_PT2_X 0.000000  REG_PT2_Y 0.000000
SUB_ORG_X 0.000000  SUB_ORG_Y 0.000000

#  Vertices ...
V    3.450   -0.422
V    3.486   -0.424
V    2.814   -2.928
V    0.510    3.646
V    0.514    3.674
V    ...
V    ...
#  550 Vertices

#
#  Point data ...
#     TYPE    SIZE   COLOR   VERTEX
P7       1         2         1
P7       1         2         2
P1       1         5         118
P    ...
P    ...
#  118 Points

#
#  Contour/Line data ...
#    TYPE   COLOR   WIDTH    VERTICES
L1        1         1          119 120 121 122 123 124 125 126\
127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144\
145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162\
163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180\
 181 182 183 184 185 186 187 188 189
L    ...
#  2 Contours
```

Comments begin with a sharp sign (#). Header data consist of keyword identifiers and associated values. In the example above, "**NVERTICES**" is a keyword for the number of vertex entries in the file and 550 is its value. **NPOINTS** and **NCONTOURS** give the numbers of point and contour elements.

Vertex entries consist of one line with a "**V**" character in column 1, a space or tab character, X-value, space or tab character, and the Y-value. The index of the first vertex entry is 1. Vertex data are in units of millimeters.

Point elements are one line entries with a "**P**" character in column 1, and space separated values of the symbol type, color, symbol size and vertex index (base 1).

Line or contour entries begin with an "**L**" character in column 1, followed by the line type, color, line width and a list of vertex indices. A back slash character continues the entry on the next line.

Study the C-language module **objfile** for methods of reading and writing **MDPlot** data files in binary and object formats.

APPENDIX D: PLOTTER/PRINTER CABLE CONFIGURATION

Digital Plotters

A serial I/O cable (not supplied) is used to connect a PC/AT or compatible AT-type computer to your digital plotter. These computers use a "D-Type" 9-pin male connector on the serial port output. The **MDPlot** program (in the function **device.c**) uses a method of serial communications called "hardwire handshake" in which control lines between the plotter and computer signal when the plotter can or cannot accept more data.

If you have a PC or PC/XT machine the serial output is from a "D-Type" 25-pin connector. You will need to purchase an appropriate serial cable for the XT that implements hardwire handshake connections. Such a cable can be purchased from most computer supply houses.

HPGL/2 Compatible Color Printers

In lieu of a digital plotter you can use an HPGL/2 compatible color printer as your plotting device. Such printers include the Hewlett-Packard DeskJet 1200C/PS and 1600C/PS. The printer must be connected to the standard printer port on your PC which is usually LPT1. A proper 25 wire printer cable (not supplied) must be used to support the HP 1200C/PS and 1600C/PS. Such cables can be purchased at most computer supply houses.