Tips and Techniques for Data Interface to DSG/3000

Jason M. Goertz
Systems Engineer
Hewlett-Packard
Bellevue, Washington

INTRODUCTION

In the last several years, businesses have seen an increasing awareness and use of computer generated graphics. While the military, along with the auto and aircraft industries, has en- joyed the use of computer graphics since the early 1960's, it has only been recently that graphic hardware and software has been in the cost range for the small or medium sized business. This is primarily due to the drastic lowering of the cost of minicom- puters and peripherals in the past few years. So great is the awareness and visibility of computer graphics that even Webster now lists one definition of the noun graphic as "a graphic representation displayed by a computer (as a CRT)."

Hewlett-Packard has shown to be a leader in this revolution of "affordable graphics." Calcomp and other companies have been building large bed and drum plotters (at a large cost) for many years, while HP entered the market with small bed pen plotters and other graphic output devices which cost only few thousand dollors. HP's largest plotter, while not the largest on the market, is half the cost of any plotter of comparable size and performance. Today, HP's graphic output devices take one of five forms:

1. Digital 8 pen plotters, such as the 7221C and 9872C.
2. Digital thermal printer-plotters, such as the 7245.
3. Raster hardcopy devices, such as the 7310.
4. Raster CRT devices, such as the 2647, 2648, 2623 terminals.
5. Desktop computers with raster CRT output, including the 9845C, with color CRT graphics. We will not deal with this in this paper.

Digital devices are those that receive data in a digital form, such as characters sent via a modem, or HP-IB link. This is different that an analog device, which plots databased upon changing voltage, current, or some other data source whose range of values is a continuum, rather than discrete points. Raster devices are those whose graphic output is formed by a matrix of dots, either turned on or off. All HP CRT terminals use raster technology. If the reader is unfamiliar with these terms, a very good computer graphics "primer" is available from Hewlett-Packard, called Becoming Comfortable with Computer Graphics. This is a small document published by HP's San Diego Division.

Computer hardware is, of course, useless without software to drive it. Hewlett-Packard's graphic software offerings are many and varied, with software available on all of the computers built. HP has even written it's own graphics languages, such as HP-GL, which most of the 8 pen plotters use as an internal language, and AGL, a high level language which is standard among many of HP's computers. HP has used these languages to write several applications, such as the graphics package on the HP1000 ("BRUNO"), and a package, aimed at the business market, which runs on the HP3000, called Decision Support Graphics, or DSG/3000.

DSG/3000 is a high level application software package. It is optimized to easily define and generate graphic plots that are commonly used in the business world. These plots are line charts, scattergrams, bar charts and pie charts. DSG is also capable of producing slides plots with only textual data, commonly referred to as "slides." DSG is not capable of performing so-called "technical" graphics functions, such as computer mapping, Computer Aided Design, or real-time graphics. However, the functions that DSG is designed to perform it performs very well, and is easy to learn to use.

The primary advantage of DSG is that it runs on the HP3000, and thus has access to all of the data storage capabilities of the HP3000. This is ideal for business graphics, which usually deals with sales figures, forecasts, budgets, and other data that a business would normally store in a computer. The purpose of this paper is to show how DSG can be interfaced to the data storage technologies available on the HP3000, and thus utilize the full potential DSG to help a business's managers make decisions. While this paper is not intended to be a primer on the use of DSG, a few of the functional aspects of the package will necessarily have to be mentioned. For those who would like to learn more about the actual use of DSG, the reference manual is excellent, and a self study course is available.

DSG'S VERSION OF THE GRAPHICS WORLD

DSG defines everything it does in terms of a chart. Charts are contained in a file called a Chartfile. It has a unique file type, and is known to MPE as the file type GRAPH, and file code 1083. This is directly analogous to the way VPLUS/3000 deals with forms within a forms file. The program GRAPH.PUB.SYS is used to define all of the characteristics of the charts within the file. These characteristics are all entered via menus (screens), and include things such as what type of chart it is (bar, line, or pie), what color pens will be used to draw the chart, how the axes will be labeled and scaled, etc. One other characteristic that is associated with the chart is the name of the data file that will be used in the plot. Currently, the only means of data input into the plot is via this file.

In addition to defining the name of the file, we must also describe the contents of the file. Both of these functions are done with the Data Definition Menu in GRAPH. In this menu, we define what the variable names are (to be used in another menu), what columns (or field number) of the file the data is in, and whether the data is numeric or textual. Up to eight variables can be defined on this menu. Every type of chart (bar, line, etc) has a specific menu associated with it. It is on this menu that the specific variables that are to be plotted are defined. Only those that are to be plotted need be defined, even though the file may contain more. This implies that one data file can be used for multiple charts. All that has to be done is define different data items to be plotted, but the same data file name.

The question at this point is how do we put data into this file? Really, this is the issue that this paper deals with. There are as many ways to create this file as there are programs and programmers. We will deal with a few of the basic ones.

First, the GRAPH itself allows the user to build a file of data. In fact, this is in some ways the most convenient way to build the file, since GRAPH builds what HP calls a self-describing file, a file type that will be used more and more in the future. This is a type of file in which the file itself contains the informaton just mentioned, that being what the variable names are, what type they are, and where in the record they are located. These data are contained in the user labels of the file, and are defined by using the Data Prompt Menu of GRAPH. This allows the menu screen to automatically define the data items to be plotted as soon as the data file name is keyed in. While this may seem the best way to define the file, there are a few drawbacks. First, all the data must be entered into the GRAPH screen. Therefore, if the data resides in another file structure, the data must be listed and keyed by hand. Only one screen of data can be entered, thus limiting the number of data points to twelve. In addition, only five variables can be en- tered on this screen. If the chart is going to be used repeatedly, a lot of time will have to be spent keying the data into GRAPH. This menu is very useful for graphs in which there is a small amount of data, the data is not resident anywhere else in the computer, and the graph is going to be used only once or twice.

For charts that are going to be used only once or twice, but requires more than twelve data points or five variables, the next best method for building the data file is to use EDITOR. HP's editor can be used, or any of a multitude of other text editors and word processors available, such as QEDIT, QAD, EDIT2, TDP (LARC), HPWORD, HPSLATE, etc. While this will not build a self describing file like GRAPH, it is a cheap way to build an ASCII file of data. Every system has at least the HP EDITOR, and most have at least one other. It is only a matter of learning how to use whatever editor is desired, and build a file. With this method, the Main menu for each type of graph will have to be used to define the data items, lengths, etc.

Another software package that can be used for this function that virtually every site has is VPLUS/3000. It is a relatively simple matter to define a form with fields for the desired data. Using ENTRY, this form can be used to key the data into a Batch file, then reformatted to a DSG compatable format by using REFSPEC and REFORMAT. All of these tools are currently provided with every system as part of the Fundamental Operating Software.

Many, if not most, of the charts that are defined are used more than once. Usually, the charts are used on a periodic basis to plot data. For example, a bar chart of monthly sales for the last 12 months is generated every month with the current month's data added and the data from a year ago deleted, or a pie chart showing expenses for the last quarter. When this type of chart is used, it is often desirable to extract data directly from the files which contain them, such as an IMAGE database or KSAM file. The trick is to get the data from the native data struc- ture into Graph's sequential file in the format and record locations defined in the chart.

If the data file is stored in a sequential file or a KSAM file, and if it is all stored as ASCII characters (as opposed to binary data, such as COMP or COMP-3), then DSG can directly access the data file. All that is necessary is to define the sequential or KSAM file in the Data Definition Menu, and process the graph. However the file will, most cases, contain too much data for one plot. For instance, a sales file might contain not only a record of total sales for a product for the last month, but also a record of every sale. It is not desirable to plot every sale record, but just the summary records. This can be done by using the Data Subset Specification on the appropriate chart Main Menu, which is the same menu used to specify which variables are being plotted. Perhaps a better way is to FCOPY the records into another, smaller, sequential file. By using the ;SUBSET option of the FROM=;TO= command, the proper records can be chosen. The file that these are copies is the one defined on the Data Definition Menu.

The most common form of data storage on most HP3000's used today is the IMAGE database. Indeed, many people buy the 3000 just for this feature, and use it heavily. Since the database files are privileged, and are formatted in a very special way, one cannot extract data from them quite as simply as from a sequential or KSAM file. Either a program must be written using IMAGE intrinsics that will extract and format the data, or an existing report-writing program must be used. HP currently has three of these report-writers: QUERY, INFORM and REPORT. INFORM and REPORT are part of the 4 module system called RAPID/3000. At this writing, REPORT and INFORM are announced but not yet released by HP. Therefore, they will only be mentioned here. There are currently many others that are marketed by OEM's, third parties, and software houses. Among these are ASK, QUIZ, AQ, and REX, to name a few.

It should be mentioned that there is a contributed utility call DB2DISK that takes data from a data set and writes it, as is, to a sequential file. At this point, the data can be treated like any other sequential file. However, DB2DISK does no formatting of the data, and if any of the data is in binary format, DSG will not be able to utilize it for a plot. Also, because of the nature of an IMAGE database, desired data will reside in multi- ple data sets, and DB2DISK will not be able to combine the data in the desired fashion. Therefore, DB2DISK can be useful if all the desired plot data resides in a single data set.

The main problem with using these programs is that most are designed to write a hardcopy report. Few, if any, are designed to write to a sequential file. The trick is to get the report writer to think it is generating a printed report, when it is really writing to a normal MPE file. Fortunately, this is made very easy by the nature of the MPE file system.

Any one of the report writers mentioned above always writes its report to an entity that MPE defines as a "file." A file is a "hole" that a program either reads data from or writes data to. The program, when opening the file, asks for the file to reside on a specific hardware device, such as a terminal, line printer, or disc. After the open, it reads and writes data normally. As long as the data looks correct, the program is satisfied. It is possible to externally redefine the device on which the file resides, along with many other file characteristics such as record size and type. This is done via the MPE :FILE command. The main bit of knowledge that must be gained is the name of the file that the report writer uses for its output. Once this is known, it is a simple matter to define a :FILE equation which will redirect the output file to a disc file. This can then be used by DSG as has been discussed before.

The following is an example of how the output of QUERY would be redirected to a DSG sequential file:

```
:BUILD DSGFILE;REC=-100,5,F,ASCII;DISC=1000

:FILE QSLIST=DSGFILE,OLD;DEV=DISC

:RUN QUERY.PUB.SYS

>OUTPUT=LP
>XEQ DSGRPT    << GENERATES REPORT TO GRAPH
                  DATA FILE. >>

>EXIT
```

The main point to be noted is that the file name that QUERY uses is QSLIST. This is necessary for the :FILE equation. The file must be defined as being on Device=Disc, as QUERY will default to a line printer, and that is an OLD file. Every other report writer has file name, that would be used in the same fashion. Also, because QUERY will attempt a page break after 60 lines, the NOPAGE option must be used in the report. The reference manual for that report writer should be consulted for this type of information.

All of the above methods are, of course, dependent upon the format of the application's data structure. If the file has binary data, or if no summary records exists, then a program will have to be written to extract and summarize the data to a form that DSG can read. Since this requires a great deal more effort than using QUERY, FCOPY or some other existing utility, things can get a bit more complicated. Many shops have a degree of red tape that must be dealt with in order to get a program developed. Whether or not this type of effort will be required is a very important consideration when developing a graphics application.

TREATMENT OF DATA

Before concluding, one thing should be mentioned about data treatment. It has been said that one can prove anything with statistics, and this adage is true also in business graphics. It is very important that a business manager understand exactly what he is looking at. In other words, what a chart says can often be misinterpreted because the person looking at it does not understand what the numbers on it mean. An example of this could be a chart of net change in sales dollars. Assume that sales had been growing by 100K every month, from 1.1 Million to 1.7 Million in 6 months. The net change is what was plotted. If this data was read as sales dollars, the person seeing this plot would think that the company had only sold 100K each month, when it had been selling close to a million. This is, or course, a somewhat absurd example. But the point is that represented data must be clearly labeled and clearly understood by the intended audience, or the plot is worse than meaningless. It can be disasterous!!

CONCLUSION

We have seen that DSG can utilize the data stored on the HP3000 very well. All that is necessary is to put the data into a format that DSG can read, that format currently being a sequential file. All that is then necessary is to define the data file name and the data variable names, along with the type of data and locations within the record. With this method of data interface, DSG becomes a very powerful tool, giving managers graphic representation of what their business is doing, thus allowing decisions to be easily made.