

USING HEWLETT-PACKARD GRAPHIC LANGUAGE AS AN INTERFACE TO SAS/GRAPH



Salvatore J. Fazzino

American Consulting Group, Inc.

Abstract

Importing a graphic image from your PC to mainframe SAS/GRAPH¹ may be possible even if the graphics package used does not directly support such an interface.

Most popular PC graphics packages provide interfaces between other PC packages through image exchange files or metafiles. An metafile is a file of generic graphing commands generated by one graphics package that can be read by another graphics package and translated into the original graphic image. Examples include VDI (Virtual Device Interface)² DXF (Drawing Exchange File)³ and CGM (Computer Graphics Metafile)⁴.

The exporting graphics package and the importing graphics package must support the same metafile format. Currently SAS/GRAPH does not support most popular metafile formats (I hear version 6.0 may support CGM).

Another format which can be used for image exchange is the Hewlett Packard Graphic Language (HPGL)⁵. If we exported a graphic image in HPGL format to a file, then process that file with a SAS program that mimics an HP plotter, we would have a functional image exchange interface or metafile.

Introduction

Several months ago, one of our users asked if we could include some PC graphs, into a mainframe based reporting system. This would require the ability to import graphic images from Harvard Graphics⁶ running on PCs to our IBM⁷ mainframe for inclusion in batch generated reports. There was no direct interface between Harvard Graphics and SAS/GRAPH, so we decided to write our own.

Harvard Graphics 2.1 exports a graphic image in either VDI or HPGL format. We decided to try using HPGL as a metafile (I already knew HPGL).

The procedures for using HPGL as a graphic exchange interface or metafile are as follows:

1. Export the graphic image in HPGL format to a file.
2. Upload the HPGL file to the host system.
3. Translate the HPGL commands into SAS/GRAPH annotate commands.
4. Re-generate the graphic image.

¹ SAS and SAS/GRAPH are registered trademarks of SAS Institute, Inc.

² Freelance Plus, SuperImage and Harvard Graphics are examples of packages using the VDI format.

³ AutoCad and Generic Cadd are examples of packages using DXF.

⁴ CGM is the ANSI standard metafile format.

⁵ HP and HPGL are registered trademarks of the Hewlett Packard Co.

⁶ Harvard Graphics is a registered trademark of Software Publishing Co.

⁷ IBM is a registered trademark of International Business Machines, Inc.

Interfacing with HPGL

Hewlett Packard Graphics Language.

The Hewlett Packard Graphics Language consists of a number of commands which drive devices like the HP7550A. These commands initialize the device, set device speed, define the coordinate system to be referenced and instruct the device in the generation of a graphic image.

Each command is represented by a two letter mnemonic, followed by a set of numeric parameters.

An understanding of HPGL is essential for the coding of the translation program. All HPGL commands fall into one or more of the following groups:

- Device commands
- Window commands
- Text commands
- Addressing commands

The translation program will only need to process a small subset of the available commands. This will be illustrated later.

For a complete listing of HPGL commands, refer to the *HP INTERFACING AND PROGRAMMING MANUAL* shipped with HP plotters⁸.

Exporting the graphic image in HPGL format.

Exporting a graphic image in HPGL format may be as simple as selecting a menu option as with Harvard Graphics version 2.1.

In Harvard Graphics version 2.1, select the export option from the main menu. From the export menu, select HPGL format. Also, a graphic image can be exported in standard or high intensity mode. Standard intensity uses hardware generated characters, while high intensity mode uses software generated characters. To make the logic in the translation program simpler, select high intensity mode.

If the graphics package you are using does not directly support exporting or saving a graphic image in HPGL format, all may not be lost. With a little ingenuity and time, you should be able to capture data sent to a COM port.

Uploading the HPGL file to the host system.

There are a number of communications packages available, most, if not all, support file transfers. For example, we use the IBM 3270 Workstation Control Program version 1.1 to communicate with our host TSO environment. To upload a file, the command syntax is:

```
A > SEND A:HPGL.PLT 'HLQ.HPGL.PLT' ASCII CRLF
```

Here we are sending the PC file HPGL.PLT to the host system file HLQ.HPGL.PLT, which is a fixed block 80 character file previously allocated. We also asked for ASCII translation and carriage return/line feed translation.

Refer to the documentation for the communications package you are using for the correct procedure to perform file transfers.

Translation Program Logic

The logic behind the translation program is simple enough. Parse through the HPGL file, pick off a command and its parameters, translate that command into the appropriate SAS/GRAPH annotate function and parameters, then output the annotate observation. Repeat until all the HPGL commands in the input file have been processed. The graphic image is then re-generated using PROC GANNO.

Coding for all possible HPGL commands would make the translation program very lengthy. Fortunately, this is not necessary. In coding the translation program, we can ignore most device control commands. If we make a few aspect ratio assumptions, we can ignore window control commands, and if software generated characters were requested, we need not worry about text control commands.

Also, the authors of the graphic package most likely used only a subset of the available HPGL commands in their plotter driver program. So we can eliminate the need to code for some commands by simply examining an HPGL file and noting which commands are used. For example, some authors may choose to not use Polygon definition commands, since a polygon can be made up of a set of independent addressing commands.

After examining a number of HPGL files from Harvard Graphics, we determined that we could re-generate any Harvard Graphics image by processing the follow HPGL commands:

- SP - Select Pen. Select a pen from the carousel.
- PU - Pen Up. Pick the pen up.
- PD - Pen down. Put the pen down.
- PA - Plot Absolute. Move the pen to a specified point.

⁸ The syntax for the commands processed by the translation program can be found at the end of this paper.

The functional equivalents in the SAS/GRAPH annotate facility would be as follows:

- The select pen command effects the value of the annotate variable COLOR. It does not result in the generation of an observation.
- The pen up command changes the value of the annotate variable FUNCTION to DRAW. It can be followed by one or more sets of X,Y coordinate pairs. Each X,Y coordinate pair will result in an annotate observation being generated.
- The pen down command changes the value of the annotate variable FUNCTION to MOVE. It can be followed by one or more sets of X,Y coordinate pairs. Each X,Y coordinate pair will result in an annotate observation being generated.
- The plot absolute command is normally followed by one or more sets of X,Y coordinate pairs. Each X,Y coordinate pair will result in an annotate observation being generated.

The translation program reads each input line into a data buffer. The data buffer is scanned for selected HPGL commands, and the first command found is then moved into the command buffer. The code to process each command looks for any possible parameters and the command terminator. SAS annotate variables are set, and annotate dataset observations are generated. Then the next command is processed.

If a command or its parameters span over to the next line, a hold buffer is used. Any data in the data buffer that can not be fully processed is moved into the hold buffer. The translation program then reads the next input line into the data buffer. The command buffer and any left over data from the previous line (now in the hold buffer) are concatenated to the beginning of the data buffer. Then processing resumes.

This continues until all input lines have been processed.

The resultant annotate dataset is then processed by PROC GANNO to re-generate the graphic image.

The following is the listing of the translation program we are using to upload Harvard Graphics Charts from the PC to SAS/GRAPH on the mainframe.

```
TSO ALLOC F(PLT) DA(HPGL.PLT) SHR; RUN;
TSO ALLOC F(GPH) DA(DEMO.GRAPH5) OLD; RUN;
```

The HPGL input from Harvard Graphics is in the file with the reference PLT. The resultant graphic image will be saved in a GCAT in the file referenced by GPH.

```
DATA TEMP;
  LENGTH COMMAND $ 2
  PARM HOLD $ 80
  FUNCTION COLOR $ 8
  DATA $ 200
  TERM $ 1;

  RETAIN X Y 0 XSYS YSYS '4' POSITION '6' FUNCTION COLOR HOLD ' ';
  KEEP FUNCTION XSYS YSYS X Y POSITION COLOR ;
```

The buffers used by the translation program are defined here. The command buffer (COMMAND) is set to a length of HPGL commands, two characters. The hold buffer (HOLD) and a parameter string buffer (PARM) are both set to the same length as an input line, 80 characters. The data buffer (DATA) is set to the maximum length of SAS variables, 200 characters. The terminator variable (TERM) is set to the length of an HPGL terminator (";"), one character.

The variables FUNCTION, XSYS, YSYS, X, Y, POSITION and COLOR are the SAS annotate dataset variables which will be in the output SAS dataset TEMP.

```
INFILE PLT;
INPUT DATA $ 1-80 ;

DATA = LEFT(TRIM(LEFT(HOLD)) || TRIM(LEFT(DATA)));
```

A line of data is read into the data buffer (DATA) from the HPGL file, and any carried over information is added to the front of DATA.

```
DO WHILE (DATA ^= ' ');
  SP = INDEX(DATA, 'SP'); IF SP = 0 THEN SP = 999;
  PU = INDEX(DATA, 'PU'); IF PU = 0 THEN PU = 999;
  PD = INDEX(DATA, 'PD'); IF PD = 0 THEN PD = 999;
  PA = INDEX(DATA, 'PA'); IF PA = 0 THEN PA = 999;
```

The data buffer is scanned for the first occurrence of each of the selected HPGL commands. If the command being scanned for is found in DATA, hold its starting position. If it is not found, indicate this by setting its position to 999).

```
POS = MIN(SP, PU, PD, PA);

IF POS = 999 THEN DO;
  HOLD = ' ';
  RETURN;
END;

COMMAND = SUBSTR(DATA, POS, 2);
DATA = SUBSTR(DATA, POS+2);
```

Using the MIN function, find the position of the first selected command in the data buffer. If the data buffer contains none of the selected commands, empty the hold buffer and return to read a new line of input. If the data buffer does contain at least one of the selected commands, assign the variable COMMAND the value of the HPGL command found.

```

SELECT (COMMAND);
WHEN ('SP') DO;
  I = INDEX(DATA,',' );
  IF I > 0 THEN DO;
    PARM = SUBSTR(DATA,1,I-1);
    DATA = SUBSTR(DATA,I+1);
  END;
ELSE DO;
  HOLD = COMMAND||TRIM(LEFT(DATA));
  RETURN;
END;
SELECT (PARM);
  WHEN ('1') COLOR = 'WHITE';
  WHEN ('2') COLOR = 'RED';
  WHEN ('3') COLOR = 'GREEN';
  WHEN ('4') COLOR = 'BLUE';
  WHEN ('5') COLOR = 'CYAN';
  WHEN ('6') COLOR = 'YELLOW';
  OTHERWISE COLOR = 'BLUE';
END;
END;

```

If the command was an SP (Select Pen), scan for the terminator character. If the terminator is not found, the command has spanned over to the next input line. Move the command and whatever is left in the data buffer into the hold buffer, then return to read the next line.

If the terminator is found, the command is complete. Move everything in the data buffer, up to but not including the terminator, into the parameter buffer (PARM). Remove all characters up to and including the terminator from the data buffer.

Based on the value of PARM, set the SAS annotate variable COLOR⁹.

An SP command does not result in the generation of an observation.

```

WHEN ('PU') DO;
  FUNCTION = 'MOVE';
  TERM = SUBSTR(DATA,1,1);
  IF TERM = ';' THEN DO;
    DATA = SUBSTR(DATA,2);
  END;
  DO WHILE (TERM ^= ';');
    I1 = INDEX(DATA,',' );
    I2 = INDEXC(SUBSTR(DATA,I1+1),',,');
    IF I2 > 0 THEN DO;
      PARM = SUBSTR(DATA,1,I1+I2-1);
      TERM = SUBSTR(DATA,I1+I2,1);
      DATA = SUBSTR(DATA,I1+I2+1);
    END;
  ELSE DO;
    HOLD = COMMAND||TRIM(LEFT(DATA));
    RETURN;
  END;
  X = PUT(SCAN(PARM,1),5.) / 125;
  Y = PUT(SCAN(PARM,2),5.) / 300;
  OUTPUT;
END;
END;

```

If the command was a PU (Pen Up), set the annotate variable FUNCTION to MOVE.

Then, the next character is checked. If it is the terminator, processing continues.

⁹ The correct mapping of SAS colors to HPGL pen numbers depend on the graphics package used to generate the HPGL file.

If that first character is not a terminator, the alternate form of the PU command has been used. The following parameter string includes a set of X,Y corrdinate pairs used to address the pens next location. For each X,Y pair (there can be any number), the annotate variables X and Y are assigned¹⁸ and an observation generated. If the parameter string spans to the next line, the command and all un-processed data is moved into the hold buffer, and the next line is read.

```

WHEN ('PD') DO;
  FUNCTION = 'DRAW';
  TERM = SUBSTR(DATA,1,1);
  IF TERM = ';' THEN DO;
    DATA = SUBSTR(DATA,2);
  END;
  DO WHILE (TERM ^= ';');
    I1 = INDEX(DATA, ',');
    I2 = INDEXC(SUBSTR(DATA,I1+1), ',,');

    IF I2 > 0 THEN DO;
      PARM = SUBSTR(DATA,1,I1+I2-1);
      TERM = SUBSTR(DATA,I1+I2,1);
      DATA = SUBSTR(DATA,I1+I2+1);
    END;
    ELSE DO;
      HOLD = COMMAND||TRIM(LEFT(DATA));
      RETURN;
    END;
    X = PUT(SCAN(PARM,1),5.) / 125;
    Y = PUT(SCAN(PARM,2),5.) / 300;
    OUTPUT;
  END;
END;

```

If the command was a PD (Pen Down), set the annotate variable FUNCTION to DRAW.

The PD command and its parameters (if any) are processed the same as the PU command.

```

WHEN ('PA') DO;
  TERM = SUBSTR(DATA,1,1);
  IF TERM = ';' THEN DO;
    DATA = SUBSTR(DATA,2);
  END;
  DO WHILE (TERM ^= ';');
    I1 = INDEX(DATA, ',');
    I2 = INDEXC(SUBSTR(DATA,I1+1), ',,');
    IF I2 > 0 THEN DO;
      PARM = SUBSTR(DATA,1,I1+I2-1);
      TERM = SUBSTR(DATA,I1+I2,1);
      DATA = SUBSTR(DATA,I1+I2+1);
    END;
    ELSE DO;
      HOLD = COMMAND||TRIM(LEFT(DATA));
      RETURN;
    END;
    X = PUT(SCAN(PARM,1),5.) / 125;
    Y = PUT(SCAN(PARM,2),5.) / 300;
    OUTPUT;
  END;
END;

```

If the command was a PA (Plot Absolute), there is no change in the value of the variable FUNCTION. The X,Y coordinate pairs following the PA command move the pen to a new location. Whether this action results in a simple move or the drawing of a line depends on the execution of a previous PU or PD command.

¹⁸ I divide the X coordinate by 125 and the Y coordinate by 300. These values are used to adjust for the world coordinate system of the plotter (10000x7200) and the world corrdinate system of the display screen (80x24). $10000/80 = 125$ and $7200/24 = 300$.

The X,Y coordinate pairs are processed in the same way as described previously.

```
        OTHERWISE;
      END;
      IF INDEX(DATA,',' ) = 0 THEN DO;
        HOLD = DATA;
        DATA = ' ';
      END;
    END;
  RUN;
```

After the input HPGL file has been processed, and the resultant annotate dataset has been created, PROC GANNO is used to generate the graphic image.

```
GOPTIONS DEVICE=GDDMPCG NOCHARACTERS;
PROC GANNO ANNO=TEMP GOUT=GPH.HGUPLD;
RUN;

TSO FREE F(PLT GPH);
```

Conclusion

This interface has worked very well for us. As of this date, the Harvard Graphics images that have been uploaded have been complete, even though only four of the HPGL commands are processed.

If at sometime in the future, an HPGL file contains a command we currently do not process, that command can be added to the translation program relatively easily.

If using HPGL as a metafile is not for you, the same logic can be applied for any metafile format. We chose to use HPGL only because I already knew the language, I had the documentation, and we needed an interface fast.

I should note that true metafiles were designed to be processed by other programs and not by devices. I would expect the output of a metafile driver to be much simpler to process than an HPGL file. So, if you are starting from scratch, you might want to try writing the translation program to process a true metafile.

Salvatore J. Fazzino

American Consulting Group, Inc.
2348 Whitney Avenue Suite #2
Hamden, CT 06518

Syntax of selected HPGL commands

Select Pen	Plot Absolute
SP n;	PA X,Y (,...);
SP;	PA;
Pen Down	Pen Up
PD X,Y (,...);	PU X,Y (,...);
PD;	PU;

Where: n is the carousel slot number.
X is the horizontal coordinate
Y is the vertical coordinate

HPGL plotter file subset for reference.

```
|----|----1----|----2----|----3----|----4----|----5----|----6----|---
```

```
IN; SC-389,9610,-297,6902; LT; PU; SP2; PA453,6602; PD453,6518; PU551,6308; P  
D565,6588; PU565,6322; PD579,6602; PU579,6308; PD677,6602; PA453,6602,702,  
5591,532,5577,1434,5591,1378,5605,1350,5633,1322,5675,1308,5703,1308,  
5745,1322,5773,1378,5801; PU; PA1422,5810; PD1423,5811,1425,5816,....
```

Note -

1. The IN, SC and LT commands will be ignored by the translation program.
2. The PU and PD commands are followed by multiple sets of X,Y coordinate pairs.

Interfacing with HPGL