

Abstract

Integrating graphics into publications was a manual task, usually involving paste-up, until we were able to combine SAS/GRAPH[®] output with our text documents. We now produce quality documents (management reports, newsletters, papers, and documentation) using IBM's Document Composition Facility (DCF) and the IBM 3800 model 3 laser printer. The key to integration lies in the use of Page Segments, which can easily be produced using SAS/GRAPH once you know how. This paper will present the techniques involved in a comprehensive document, which should eliminate most of the work involved in gleaning the necessary information from the multitude of manuals provided by IBM and SAS Institute.

Introduction

When our site began planning for the installation of our first 3800 model 3 laser printer, we were very excited by the graphics capabilities of the device. The advantages of combining text with graphics for reporting and documentation are evident. We soon found that the key was to have SAS/GRAPH put the output into IBM Page Segment format. The problem was finding out how to accomplish this. After poring through several IBM manuals and SAS[®] manuals, and making some calls to SAS Institute, the problem was solved. The process is not difficult, just undocumented. This paper will show you how to produce Page Segment output and give you some hints on inserting your graphics into a DCF text document.

Prerequisites

To successfully employ the techniques presented in this paper, you will need the following hardware and software:

- SAS Base and SAS/GRAPH software, Version 5 or higher.
- Graphical Data Display Manager (GDDM), Rel. 4 or higher.
- Document Composition Facility (DCF), Rel. 3 or higher.
- One of the following environments for DCF that support Page Segments:
 - TSO
 - CMS
 - ATMS-III (Advanced Text Management System)
- One of the following printers:
 - IBM 3800-3
 - IBM 3820
 - IBM 4250
- One of the following printer drivers:
 - PSF (Printer Support Facility for 38xx)
 - CDPF (Composed Document Printing Facility for 4250)

Our environment includes TSO, PSF, and 3800-3 printers; but the same methods should work for the other options listed. You may have to make some changes for your environment.

Document Composition Facility

DCF consists of two component parts, SCRIPT/VS and GML (Generalized Markup Language). SCRIPT/VS is the text processing program. It processes a file of text and typesetting commands (*markup* or *control words*) to produce a page output file for the printer being used. If only SCRIPT/VS was used, you would need control words to specify each detail of the document. GML functions as a macro facility for SCRIPT/VS. GML markup consists of *tags* that result in several SCRIPT/VS commands being processed. GML comes with a starter set document profile designed for general documents. Many IBM manuals are done using this facility, so the appearance should be familiar to you.

The SCRIPT/VS command used to include graphics in the text stream is the *Segment Include* (SI) command. The graphic must be stored in a *Page Segment* library, such as "SYS1.PSEGLIB". Usually the page segment will be included in a Figure, but this is not a requirement. You may also overlay one segment over another. Details for doing this are in the *DCF SCRIPT/VS Language Reference* (SH35-0070). In order to use this facility, you must get your SAS/GRAPH output stored in a Page Segment library. The following section will tell you how to do this.

Page Segments

Several factors should be considered when using Page Segments to store and display graphical data. They include:

- Physical storage size on DASD.
- Size of the graphic on the printed page.
- Placement of the graphic in the figure.
- Orientation of the graphic on the page.
- Specifying the Page Segment library.

This section will discuss these factors.

Physical storage size on DASD can cause problems if the space available to you is limited or you do not have update capability to the production page segment library (SYS1.PSEGLIB at most installations). The resolution on the 38xx printers is 240x240 pels per square inch, or 57,600 pels. The 4250 resolution is higher and requires more storage. GDDM will divide graphics into *cells* (each cell is 32x32 pels) which are only output if at least one pel in the cell is "on"; but this will not always save you much space. You may have to delete your graphics after your document is printed. This will depend on the storage limitations of your installation. If you are using your own page segment library, it must either be online to PSF when your document is printed or you must specify the *INLINE* parameter on the segment include command. The *INLINE* parameter will tell SCRIPT/VS to include the graphic in the output text stream.

Size of the graphic on the printed page can affect your SAS jobs. You will need to use a GDDM nickname (see "Interfacing SAS/GRAPH with GDDM") that will produce a graphic of the desired size for the figure. You must also proportion the graphic for the space to be used. We have obtained the best results by having SAS create the original graphics in the desired size.

Placement of the graphic in the figure will influence the proportions used to build it. The origin of a graphic is placed at the upper left hand corner of a figure, relative to the direction of movement of paper through the printer (i.e. the absolute top of the paper, not the top of the page). The positive Y axis points down the page. The graphic is *not* centered in the figure, although you may employ native SCRIPT/VS commands to position it.

Orientation of the graphic on the page must be considered when running your SAS job. Page Segments are not rotated by PSF. If your

output is going to a DCF logical device that produces a rotated page, you must get SAS/GRAPH to produce a rotated graphic. One way to do this is with the Template facility of SAS/GRAPH. There is a standard naming convention within Page Segment libraries which will help you keep track of the orientation of your segments. Segments are named SxNAME, where 'x' indicates the amount of rotation. 1 is no rotation, 2 is -90 degrees, etc. NAME will be the name you assign. This is useful if you need to use the same graphic in reports printing in different orientations. A similar convention is used for IBM supplied character sets, which are also named by orientation. For example, "X1" tells you that a character set is not rotated.

Specifying the Page Segment library is not as easy as it would seem from the DCF documentation. The SEGLIB option allows you to name your private page segment library for use by DCF. However, if you do not specify the INLINE option on the Segment Include SCRIPT/VS command, your private library must be concatenated behind the production Page Segment library in your installation's JES startup procedure in order to be accessed. Specifying the INLINE option will greatly increase the size of the output datastream from DCF. This could cause problems if your installation is short on spool space.

It is also important to note that printing graphics on the 38xx printers can cause reduced throughput. This may cause you problems if someone notices.

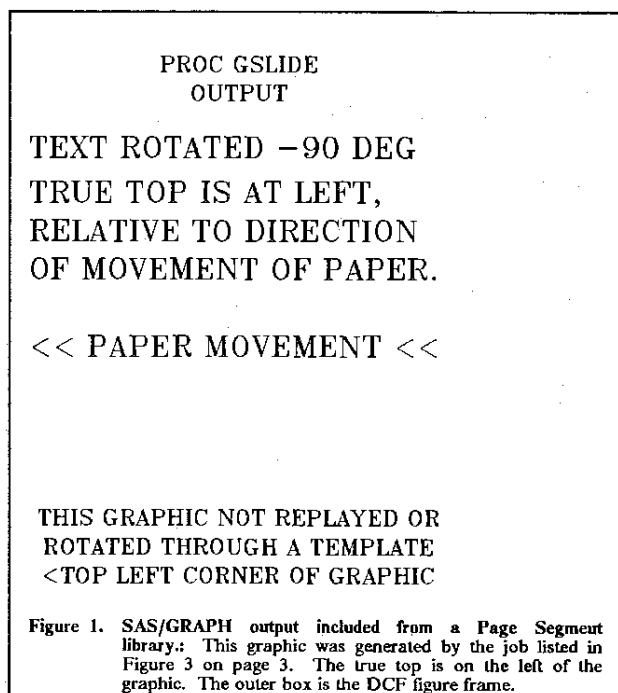


Figure 1. SAS/GRAPH output included from a Page Segment library. This graphic was generated by the job listed in Figure 3 on page 3. The true top is on the left of the graphic. The outer box is the DCF figure frame.

Interfacing SAS/GRAPH with GDDM

Several topics need to be discussed when using SAS/GRAPH and GDDM to display graphical data. They include:

- GDDM Nicknames and Tokens.
- The SAS/GRAPH GOPTIONS statement.
- Use of the ADMDEFS file.

This section will discuss these topics.

GDDM Nicknames and Tokens refer to entries in an external dataset used by GDDM to determine device characteristics. If your installation is using GDDM, there should be an entry in the standard SAS

JCL procedure with a DDname of ADMDEFS. For the printers used in this article, which are GDDM Family 4 devices, both a nickname and a token are required. The GDDMFAM4 device driver will be used. The Token used must specify a Family 4 printer. The Nickname name will also be the DDname of your output file. Nicknames will be discussed further with the ADMDEFS file below.

The SAS/GRAPH GOPTIONS statement should specify the Device, Nickname, and Token to be used when calling GDDM. An example follows:

```
GOPTIONS DEVICE=GDDMFAM4
GDDMNICNAME=IBMPSEG
GDDMTOKEN=SASPLOTT DISPLAY
HSIZE=10.5 VSIZE=7.0;
```

The Token name will normally be your installation's standard for the printer that you are using. The Nickname will probably be defined in your ADMDEFS file (see below), but your installation may define standards that meet your needs for Page Segments.

Use of the ADMDEFS file is the easiest way to tailor your Page Segment output to exactly fit your needs. Your primary need will be to be able to specify the size of the Page Segment being built. You can do this via the HRIPSIZE parameter of a Nickname that you define in source file format in the ADMDEFS file. See Figure 3 on page 3 for an example of a source format ADMDEFS file included in the JCL. It would also be possible for your installation to predefine standard size Page Segments in the installation ADMDEFS file. If your installation defined standard Nicknames for quarter, half, and full page graphics, you would not need your own ADMDEFS file. The sizes specified in your HRIPSIZE parameter and in your SAS/GRAPH HSIZE and VSIZE (in the GOPTIONS) should match.

The key to obtaining Page Segment output is also in the ADMDEFS file. As one of the processing options, you must define:

```
PROCOPT=((CDPFTYPE,SEC))
```

This tells GDDM that the output is for a Page Segment. (Composed Document Printing Facility Type is secondary.) A CDPFTYPE of primary indicates that output is going directly to the device.

Interfacing Page Segments with DCF

The sample run in Figure 3 on page 3 will produce a Page Segment with a horizontal size of 4.0 inches and a vertical size of 3.0 inches in file IBMPSEG. The file could be a Page Segment library member or a sequential file. This example was used to generate Figure 1. Note that the horizontal size is larger in the Nickname, but the vertical size is larger on the page. This is due to the way that PSF handles page segments.

Page segments are not rotated when they are printed, even if your page is being rotated. You must use SAS/GRAPH to orient them correctly for your application at the time that they are created. This paper was composed using DCF on paper having the larger dimension in the horizontal direction. This is sometimes referred to as "landscape orientation." DCF was told to rotate the page ninety degrees via the device specification parameter, which includes page size and orientation information. DCF rotates in what appears to be the clockwise direction for positive rotation (counterclockwise is used by SAS/GRAPH and most cartesian geometry.) This is due to the fact that the positive Y axis is considered to point downward, based on the direction of movement of the paper through the printer. To compensate for this, you must tell SAS/GRAPH to rotate your graphics minus ninety degrees as in Figure 3 on page 3.

Including Page Segments in DCF Documents

Page segments are included in documents by invoking the SCRIPT/VS segment include command, ".si". We normally include graphics within figures, but this is not a requirement. See Figure 2 on page 3 for the coding used to create Figure 1. Remember that DCF will not change either the size or orientation of your page segment. The absolute upper left-hand corner of your graphic will be placed at the current location on the page.

As discussed previously, if your page segment library will not be online to PSF at the time your document is printed, you need to do two things:

1. Use the `INLINE` parameter on the segment include command to tell DCF to include the data required to print your page segment in the output data stream.
2. Use the `SEGLIB` parameter on the `SCRIPT/VS` command line to give `SCRIPT/VS` the name of your page segment library.

If your graphics have not yet been saved in a page segment library, you can specify width and depth parameters on the segment include command. These values are ignored if the actual page segment is found in the page segment library. This technique permits you to produce all drafts except the final one without having your page segment library online; but with the correct space being left for the graphics.

```
:fig id = Fpseg1 width = column place = inline frame = box
.si S2JIM1 inline
:figcap.SAS/GRAPH output included from a Page Segment library.
:figdesc.This graphic was generated by the job listed in
:figref.refid = Fjcl1..
The true top is on the left of the graphic. The outer box is the DCF
figure frame. This is an example of a Figure reference. See :figref
refid = Fig6. for the coding used to generate this text.
:efig.
```

Figure 2. Including page segments.: The above text and DCF coding was used to generate Figure 1 on page 2. The actual segment include is performed by the ".si" command.

Acknowledgment

I would like to thank Martin Mincey of SAS Institute for his assistance in helping us to find the solution to the Page Segment problem.

References

The following publications will be helpful to those using the techniques described in this paper:

- *SAS/GRAPH User's Guide: Version 5 Edition*, SAS Institute.
- *GDDM Installation and System Management*, IBM (SC33-0152).
- *DCF GML Starter Set User's Guide*, IBM (SH20-9186).
- *DCF SCRIPT/VS Language Reference*, IBM (SH35-0070).
- *DCF SCRIPT/VS Text Programmers Guide*, IBM (SH35-0069).
- *In House Publishing Primer*, IBM (GG24-1717).

Author

James W. Mathews
 American Express Travel Related Services Company
 Travelers Cheque Operations Center
 4315 South 2700 West
 Salt Lake City, Utah 84184-1600
 (801)965-2206

SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc., Cary, NC, USA.

```
//TSTSJMS JOB TCTSV,'MATHEWS',CLASS = R,MSGCLASS = T,NOTIFY = TSTSJM
/*JOBPARM L = 100,T = 15,R = 1600
//STEP1 EXEC SAS,REGION = 5000K
//ADMDEFS DD * *** INLINE ADMDEFS FILE FOR NICKNAME DEF'N ***
NICKNAME REPLACE,NAME = IBMPSEG,DEVTOK = SASPLOT,FAM = 4,
PROCOPT = ((CDPFTYPE,SEC),(HRIFORMT,CDPF),(HRISPILL,NO),
(HRISWATH,10),(HRIPSIZE,40,30,TENTHS))
//OUT1 OUTPUT CHARS = GT20,PAGEDEF = 090C0
//IBMPSEG DD DSN = TSTSJM.GDDM.GRAFOUT,DISP = SHR
/*IBMPSEG DD DSN = TSTSJM.GDDM.GRAFOUT,
/* SPACE = (CYL,(5,1),RLSE),UNIT = SYSDA,
/* DISP = (NEW,CATLG,DELETE)
//SYSPRINT DD SYSOUT = *
//SYSIN DD *
* PAGE SEGMENT DEMONSTRATION;
OPTIONS NOTEXT82;
GOPTIONS DEVICE = GDDMFAM4 GDDMNICKNAME = IBMPSEG GDDMTOKEN = SASPLOT
HSIZE = 4.0 YSIZE = 3.0 NOTEXT82 DISPLAY GUNIT = PCT;
PROC GSLIDE GOUT = SAVEGRAF BORDER;
TITLE1 H = 6 A = -90 ' ';
TITLE2 F = TRIPLEX H = 6 C = BL A = -90 'PROC GSLIDE';
TITLE3 F = TRIPLEX H = 6 C = BL A = -90 'OUTPUT';
TITLE4 H = 4 A = -90 ' ';
NOTE F = TRIPLEX H = 8 C = BL A = -90 'TEXT ROTATED -90 DEGREES';
NOTE H = 2 A = -90 ' ';
NOTE F = TRIPLEX H = 8 C = BL A = -90 'TRUE TOP IS AT LEFT';
NOTE F = TRIPLEX H = 8 C = BL A = -90 'RELATIVE TO DIRECTION OF';
NOTE F = TRIPLEX H = 8 C = BL A = -90 'MOVEMENT OF PAPER.';
FOOTNOTE1 F = TRIPLEX H = 6 C = BL A = -90 'THIS GRAPHIC NOT REPLAYED OR';
FOOTNOTE2 F = TRIPLEX H = 6 C = BL A = -90 'ROTATED THROUGH A TEMPLATE';
```

Figure 3. JCL and SAS statements to generate Figure 1 on page 2.: Note the inline ADMDEFS file, a sequential file or a PDS member could also be used.