

Conjoining Images and SAS Graphics with DATA Step Graphics Interface (DSGI) Navigator

Eric T. Sun, C. R. Bard Incorporation, Murray Hill, New Jersey

ABSTRACT

The DATA Step Graphics Interface provides the graphics routines used by SAS/GRAPH software interactively joined with datasets. SAS programmers can enjoy the 'smooth-sailing' or freedom of creating custom graphs in a DATA step programming style. However, just like sailing, there is no freedom without directions. To start a graphic job by selecting a function call such as 'draw line' is much easier than deciding where and how to put it. DSGI Navigator, a SAS/AF[®] Frame Entry application runs with SAS version 6.11 in MS Windows 95, was then designed to ease the task of preparation and maximize the freedom of using DSGI by utilizing the GUI features provided by object classes such as Image and SAS/GRAPH[®] Output. This application currently has three functions, form reproduce, data mapping and graph join. It also produces DSGI source codes after finishing a function through its DSGI program organizer which makes the codes reusable and portable or smoothes the learning steps for novices to DSGI.

INTRODUCTION

DSGI gives you all the powerful functions and call routines which follow the standards of Graphics Kernal System (GKS). The most outstanding feature of DSGI is directly performing graphics generation with the flexibility of a DATA step. It has some similar function concepts that are offered by SAS/GRAPH annotation facility and both retrieve data values into graphic function calls in DATA steps. But annotation facility takes one more step, PROC, to fulfill a sub-graph with the graphical setup from a main graph. With DSGI, you start with a blank work sheet and do whatever you want. But here comes the problem, without the natural constraints of SAS/GRAPH, how do you start a DSGI task? How do you transform image objects into SAS/GRAPH catalog entries through the magic of DSGI? What is the best way to manipulate graph entries using DSGI as a console? What other potential graphic functions can DSGI do in the real world? What is the strategical method to benefit people who want to learn DSGI? It is the unknown that blocks people from realizing the

power of DSGI even though it can be found in the SAS/GRAPH menu. It is also the simplicity of DSGI that ironically forces some people to give up after several tedious tries of pinpointing x/y values. DSGI Navigator is developed to enhance the 'direction' and ease of programming with DSGI.

THE DSGI Navigator APPLICATION

The DSGI Navigator, a SAS/AF[®] Frame Entry based application, employs the most advanced GUI and system resources from SAS version 6.11. It is designed for both real world functionality and possibly a learning tools for people interested in DSGI programming. With work experience in window environment and basic SAS programming knowledge, plus hypertext on line helps, graphs and related DSGI source codes can be created and stored into specific project directory or library. Various graph saving options were considered to increase the utilization of output graphics with other commercial softwares or word processing packages.

Primary Design Goals

- There is no intention to replace any one of SAS/GRAPH procedures with DSGI Navigator at the very first design stage. Even some functions in this application may be found in existing procedures with similarities, but the approaches and results are different.
- A GUI based tool set shall provide a pure visual DSGI programming environment, that is, everything from drawing routines to the place to anchor graphics can be seen and selected or 'navigated' all the time. Based on the pre-defined functions in DSGI Navigator, any single DSGI programming code shall be done through a point and click GUI fashion. Global DSGI source code structure has been modulated but can be customized with adequate DSGI knowledge.
- Either the output graphics or the derived source codes can be saved and reused. The output graphic can be saved in SAS/GRAPH catalog entries, external image files or to the clipboard from which cut/paste can be performed and several industry standard output file formats can be used.

users can get the DSGI program stored in the background and if necessary, modify the source codes through the Program Organizer. Program editor can be setup to the one users preference. Dynamically users can view the current work right away to ensure the precision of output. or the work partially done last time before start it again.

2 Data Mapping

After a data form is successfully converted to a SAS graph created by DSGI program, the data source shall be converted to or entered as a SAS dataset so the further representation of business forms and information can be fulfilled completely in SAS system. Any format or informat associated with a data variable shall be carefully created and defined to reflect the true layout of that variable in a form. When the Data Mapping function is activated, the target form image is brought into the workspace first, the prepared dataset is selected (Figure 3).

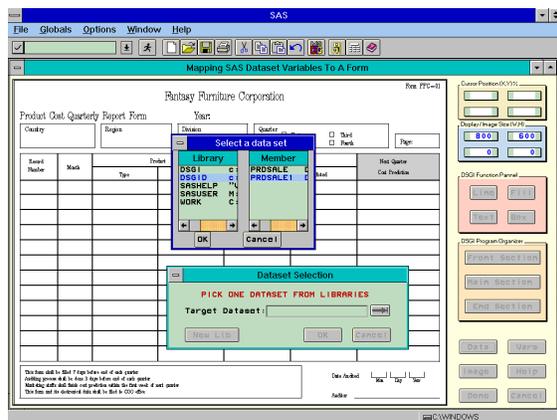


Figure 3 - Data Mapping at dataset selection

Its contents, such as type, length, format and label are displayed in an Extended Table Entry. (Figure 4) With a variable selected, the spot to anchor it in a data field and the suitable font size to display, it can be determined quite easily by referencing an image font size option window. Proper program logics shall be added into the main DSGI program in DSGI Program Organizer to ensure correct display information, such as numerical variable SEX may have to be mapped back to Male/Female check boxes on a form. Most cases like this can be easily done by users who are familiar with DATA_NULL_ programming. To ease the data variables mapping steps, an option to reuse the row or column position of previous mapped variable can

line up group of variables in certain data collection sequence.

Since this data mapping function can determine how/where/what/when to present data related to a form, It can be quite useful to present data in various layouts or patterns. For example, in real business world, certain information may be reviewed with corporation internal setup. Market price prediction in a form can be blocked according to reviewer's department, title and function. Likewise data management control information shall only seen by data processing staffs. Time released information can be scheduled base on system clock. So a dataset associated a form can include more information than those collected from a form and be presented in business nature.

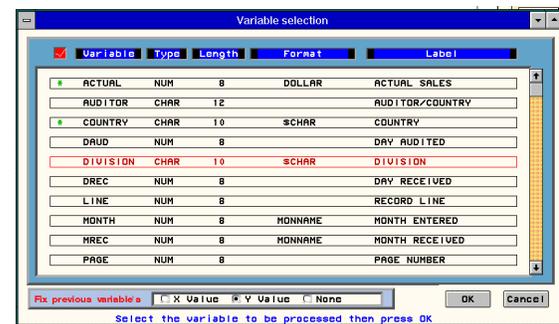


Figure 4 - Data Mapping at variable selections

3 Join Graphs

Graphs created either by SAS/GRAPH procedures or DSGI can be grouped and saved into graph catalogs according to their business purposes. There are several ways to generate a graph which contains all the possible information in graphical format, i.e. double vertical axes plot, multiple variables plot, or even a mix of these two types of graphs with descriptive annotations. For a page layout with multiple graphs, PROC GREPLAY is right there to serve this purpose with SAS default or user defined template options.

The Join Graphs function comes with a totally different idea to produce a combo graph from a SAS/GRAPH catalog. First, it provides direct edit on a graph through either built in DSGI drawing buttons or SAS/GRAPH editor. Second, since a GUI frame entry is used to select and preview graphs in advance, the way to join graphs can be decided visually in a free form style. In this frame (Figure 5), there are two graph display areas with their own x/y reference system.

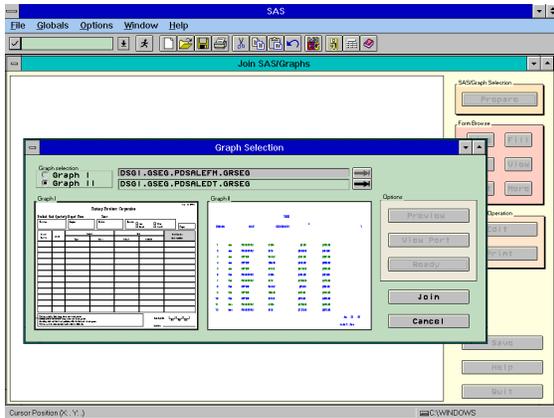


Figure 5 - Join Graphs at graph preparations

The within or cross over reference options grant users a free-form resizing and overlapping of two graphs. As a result at 1 to 1 ratio (Figure 6),

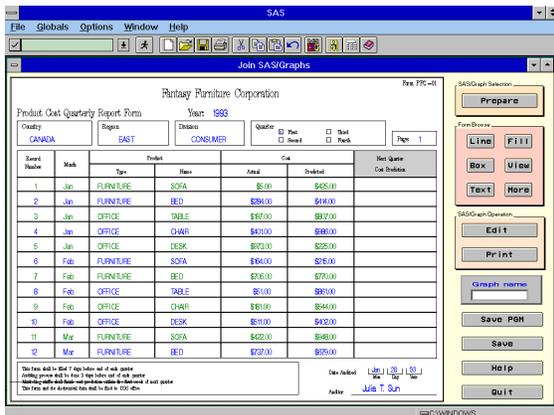


Figure 6 - Join Graphs with overlapped two graphs

it can overlap graphs like PROC GREPLAY does, but it does not need any template at all to produce a new graph. This is a demonstration that represents SAS dataset in original data collection form described in previous two functions. The advantages of this form data presentation are more than just producing on screen data review tools. The efficiency of program execution and output generation was one of major triggers to develop this DSGI application. To accomplish overall efficiency, static graphs are only generated once and reused as 'wallpaper'. The rest graphs generated dynamically are relatively simpler and smaller compared to the wallpaper. The static graph, like a business form, can be saved as a graph entry in a catalog or preprinted first so related data graphs can be over printed on it later.

A joined graph with any visually decided ratio can also be easily created. This is done through the ViewPort Allocation options (Figure 7 upper right).

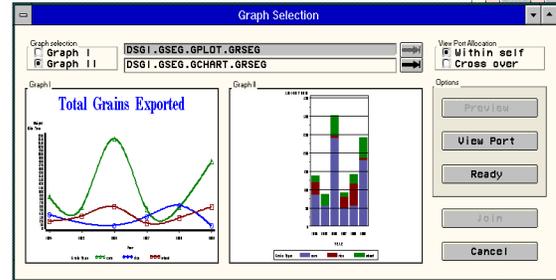


Figure 7 - A Plot and a Bar Chart before visually join

with two simple point and clicks and the joined graph (Figure 8) can be determined whether acceptable and/or saved right away into current catalog or clipboard.

Even more, the DSGI source code is always ready for reuse. Noteworthy, if there is a well defined SAS graph catalog structure in a shareable platform such as a client server system. Instead of sending an image or graph file to a remote partner, just e-mail the tiny ASCII program file to execute on the remote site. It can be a big plus to time savings. The source code generated by DSGI Navigator for the example (Figure 6) is listed below:

```

/* DSGI program used to join SAS/Graphs in the
   same graph catalog */
/* Programmer ID: SUNERIC
   Date/Time: 27AUG96 08:47 */
data _null_;
rc=gset('catalog','dsgi','gseg');
rc=ginit();
rc=graph('delete','joinhere');
rc=graph('clear','joinhere');

**** graph1 drawing;
rc=gset('viewport',1,0,0,1,1);
rc=gset('window',1,0,0,100,100);
rc=gset('transno',1);
rc=graph('insert','PDSALEFM');

**** graph2 drawing;
rc=gset('viewport',2,0,0,1,1);
rc=gset('window',2,0,0,100,100);
rc=gset('transno',2);
rc=graph('insert','PDSALEDT');

rc=graph('update','noshow');
rc=gterm();
run;

```

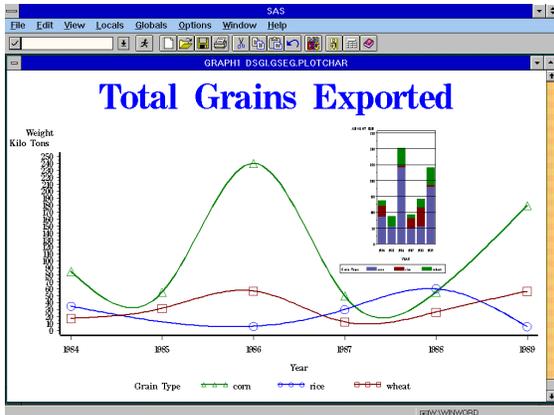


Figure 8 - A joined graph without using Templates

FUTURE EXPANSION

DSGI Navigator exhibits its unique strategy to take advantage of using DSGI programming. Besides current established functions, there are some other possible expansions:

Enhanced Data Browser

This function will not only browse a SAS dataset in its original form with proper page selection, but combined with data base validation, data fields in question or errors can be printed on pages of pre-printed forms. Graphic attributes can be changed (i.e., shaded or text color changed to red) so data collectors at different sites can easily spot the problem data.

SAS/GRAPH Catalog Supervisor

With the on-line visualization capabilities, DSGI Navigator may transform a new application which can supervise and organize SAS graphs and image files created by other commercial softwares. Functions can be designed like browse, copy, manage or manipulate SAS graph within or across catalog entries or even joined with PROC GREPLAY/GPLOT etc. to mix flexibility with conventional SAS graph creations.

WHAT SAS MAY HELP

As a GUI tools, SAS/AF and Screen Control Language really demonstrate their capabilities to satisfy various business needs. There are a lot of frame entries that deal with library, file, and graph

were found handy in AF system resources and adapted in the DSGI Navigator. Unfortunately there are no documentation or open codes at all for fine tunes or further customization. It is understandable at this early stage a lot of programming resources maybe still needs some improvements or well documentation. An open environment of AF toolkits design and implementation like Visual Basic® will definitely benefit all parties in exploring the power of SAS/AF.

CONCLUSION

DSGI Navigator reveals a more human nature way to produce graphs with DSGI and handle graphs more straight forward through eyes. Its DSGI codes automation feature leads a new sequence of creating graphs before writing codes which can be executed to regenerate the same graphs remotely.

For more information regarding DSGI Navigator please contact

Eric T. Sun
 C. R. Bard Incorporation
 730 Central Avenue
 Murray Hill, NJ 07974
 Voice: (908) 277-8114
 e-mail: eric.sun@crbard.com

ACKNOWLEDGMENTS

I would like to thank Harvey Monder for his DSGI Publications and direct guidance and opinions to this application. Also I wish to thank Fei Tan's third-party challenges to the function ideas and suggestions during the development. A sincere thank is also due Teresa Cuchiaro for her reviewing and edit checking of this paper.

REFERENCE

Monder, Harvey (1995), "Creating Tables Using the Data Step Graphics Interface (DSGI)," Observations, Third Quarter, 1995.

SAS, SAS/AF, SAS/GRAPH are registered trademarks or trademarks of SAS Institute, Inc. in the USA and other Countries. © indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.