

A COMPARISON OF PROC GREPLAY AND THE DATA STEP GRAPHICS INTERFACE (DSGI)

Lynn M Thome-Polingo, G.D. Searle

ABSTRACT:

This paper will show the ability to surpass the realm of PROC GREPLAY and to end with the unexpected ability to create customized diagrams with DATA Step Graphics Interface (DSGI), to meet the standards of critical viewers. Still, GREPLAY is suited for quick and easy use through its windowing features.

The objective of this paper is to explain and illustrate the differences between these two tools. This way one knows the advantages and limitations to consider when selecting one versus the other as a tool for graphic development.

INTRODUCTION:

There exist two methods in SAS which allow the user to play back graphs stored in SAS graphics catalogs. These methods are PROC GREPLAY and the DATA Step Graphics Interface (DSGI) tool. Both methods are very similar, but DSGI has a greater capability to customize output. This paper will begin to discuss how the two methods are run, show a break down of abilities, and display various customization features.

The environment at Searle consists of a VAX cluster, using Digital Postscript printers. The work-stations consist of either a Mac or IBM PC with an emulation package connecting them to the cluster.

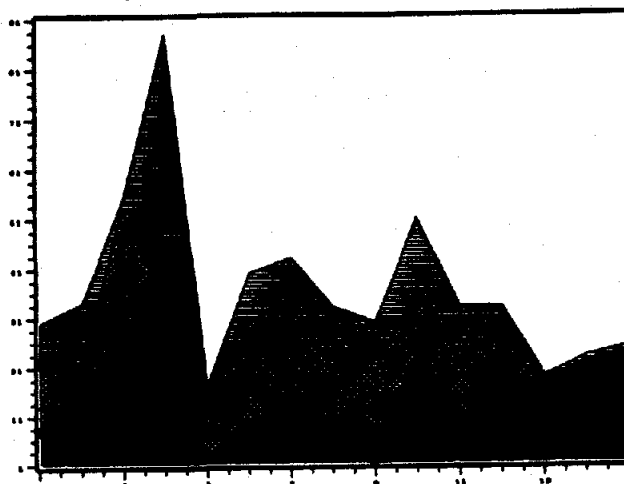
BASIC WORKINGS OF PROC GREPLAY:

Definition: This procedure replays graphics output stored in SAS catalog entries and manages the entries within the catalogs. PROC GREPLAY also creates templates and color maps that can be used while replaying the graphics output.¹ PROC GREPLAY has the ability to use up to four different catalogs at once, the catalogs are listed as follows, Input (IGOUT), Output (GOUT), Template (TC), and Color Map (CC) catalogs.

When running PROC GREPLAY, the user must remember that the procedure will continue to run until stopped. It is very important to remember to code a STOP, QUIT, or END statement with this procedure. Also when a SAS session is stopped the procedure will be stopped also. To simply demonstrate how PROC GREPLAY replays a graph, the following code was developed,

```
PROC GREPLAY IGOUT=PAPER.P2 NOFS;
  REPLAY PLOT1;
RUN; QUIT;
```

the output is shown as follows,



PROC GREPLAY, unfortunately does not have the ability to add headers and footers on its own, they must be previously designed within PROC GSLIDE or PROC GPLOT. Lets next take a look at how DSGI would handle the above task.

BASIC WORKINGS OF DSGI:

Definition: The DATA Step Graphics Interface enables the user to create graphics output within the DATA Step, or from within a SCL application. Graphic routines used by SAS/GRAPH* can be called to generate an entire custom graph or to add features to an existing graph not unlike the Annotate facility of SAS/GRAPH.² An example of DSGI code similar to the previous example for PROC GREPLAY is as follows,

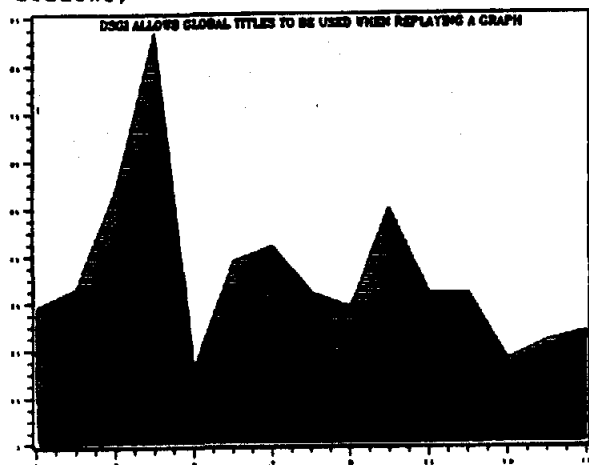
```
TITLE1 'DSGI ALLOWS GLOBAL TITLES TO BE
      USED WHEN REPLAYING A GRAPH';
```

```
DATA NULL ;
  /** SET CATALOG WHERE OUTPUT EXISTS ***/
  RC= GSET('CATALOG','PAPER','P2');

  RC= GINIT();
  RC= GRAPH('CLEAR','DSGII','Simple
    print of plot, similar to PROC
    GREPLAY');
  RC= GSET('VIEWPORT',1,15,20,90,
    .80);
  RC= GSET('WINDOW',1,15,15,95,75);
  RC= GSET('TRANSNO',1);
  RC= GRAPH('INSERT','PLOT1');
```

```
RC= GRAPH('UPDATE');
RC= GTERM();
RUN;
```

the output for the code shown is as follows,



DSGI as shown in the output has the ability to use some of SAS/GRAPH's global commands, namely Titles. Lets now look into a more advanced PROC GREPLAY run.

ABILITIES, PROC GREPLAY:

PROC GREPLAY can be either used in full-screen windowing mode, or with line or batch mode statements. The examples found in this paper will be using line or batch processing. However, the windowing feature for this procedure is very handy.

PROC GREPLAY has the ability to replay graphs and manage catalogs containing graphs. The managing features available in PROC GREPLAY help to keep the Input catalogs (IGOUT) clean and organized, and allows graphics output in a catalog to be retrieved easily. There are five tasks available for managing a catalog, Copy, Group, Delete, Modify, and Move. A more extensive description then the following can be found in SAS/GRAPH, V2, p. 1198.

```
COPY    -> copies a graphics output
           from the input catalog
           (IGOUT) to the output
           catalog (GOUT), only able
           to use if an output catalog
           (GOUT) is specified.
GROUP   -> arranges graphics output
           into logical groupings.
DELETE  -> deletes unneeded graphics
           outputs.
MODIFY  -> changes graphics output
           names and descriptions.
MOVE    -> reorders graphics output.
```

These managing features are very exciting and helpful, such as when the user is pulling catalog information for replay purposes, GREPLAY has the ability to keep the IGOUT catalog entries clean. If a catalog is written to more than once

there will exist the old and the new output entries, this can cause inappropriate entries to be printed if the user is not careful. The DELETE statement, will allow only those wanted entries to be kept after every run.

An example of why managing features are helpful could be shown when running under production mode and your graphic output must be created weekly. As a user, you are not going to want to have to clean out the catalogs yourself, separate from the running of the program. The following code contains a few of the above mentioned managing tasks.

```
FOOTNOTE1 F=SPECIAL H=2 'J J J' F-CENTX
          H=1.5 ' A LIST '
          F-WEATHER H=2 'C C C' F-CENTX
          H=1.5 ' B LIST '
          F=SPECIAL H=2 'D D D' F-CENTX
          H=1.5 ' C LIST '
          F=SPECIAL H=2 'A A A' F-CENTX
          H=1.5 ' D LIST '
TITLE1 H=2.5 F-CENTX 'PROC GREPLAY
          ABILITIES';
```

```
PROC GSLIDE GOUT=PAPER.P2 DES='TITLES FOR
GREPLAY2 PROGRAM' NAME='SLIDE1';
RUN;
```

/* The Input Catalog contains 6 plots and the first two will be deleted */

```
PROC GREPLAY NOFS;
  IGOUT=PAPER.P2;
  DELETE PLOT1 PLOT2;
  MOVE PLOT3 AFTER PLOT6;
  MOVE PLOT5 BEFORE PLOT4;
  GOUT PAPER.P2;
  TC=TEMPCAT;
  TC=TEMPCAT;
  TC=TEMPCAT;
  TC=TEMPCAT;
  TC=TEMPCAT;
  TDEF NEWFOUR DES='GREPLAY FOR SUGI 17
  PAPER'
```

```
1/LLX= 0 LLY= 0
   ULX= 0 ULY= 95
   URX= 100 URY= 95
   LRX= 100 LRY= 0
```

```
2/LLX= 5 LLY= 10
   ULX= 5 ULY= 45
   URX= 48 URY= 45
   LRX= 48 LRY= 10
```

```
3/LLX= 5 LLY= 50
   ULX= 5 ULY= 85
   URX= 48 URY= 85
   LRX= 48 LRY= 50
```

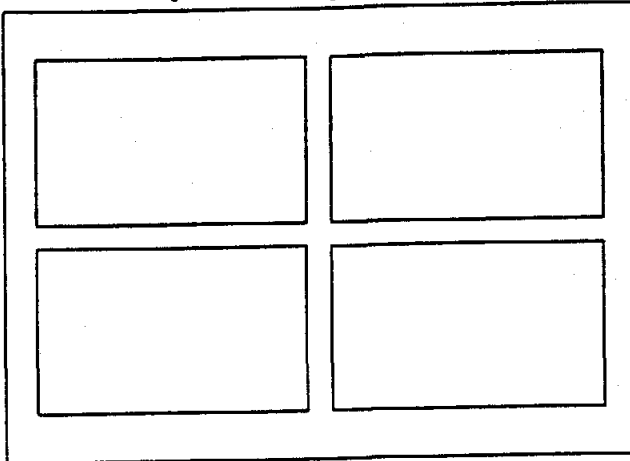
```
4/LLX= 52 LLY= 50
   ULX= 52 ULY= 85
   URX= 95 URY= 85
   LRX= 95 LRY= 50
```

```
5/LLX= 52 LLY= 10
   ULX= 52 ULY= 45
   URX= 95 URY= 45
   LRX= 95 LRY= 10;
```

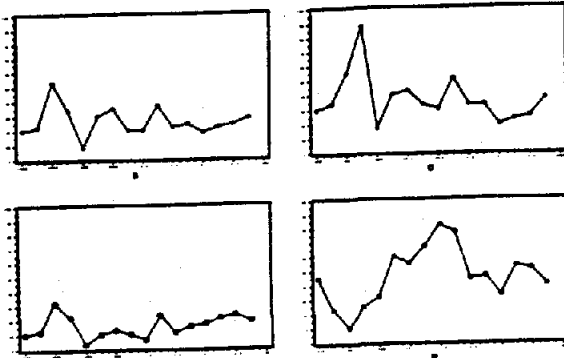
```
PREVIEW NEWFOUR;
TEMPLATE NEWFOUR;
TREPLAY 1:SLIDE1 2:PLOT3 3:PLOT4
        4:PLOT5 5:PLOT6;
```

```
RUN;
QUIT;
```

The following is the output created from the above code along with a preview of how the templates are placed.



PROC GREPLAY ABILITIES



For a more complete list and descriptions of the options available in PROC GREPLAY, see SAS/GRAPH, V2 pp. 1199 - 1222. Now lets create the same output with DSGI.

ABILITIES, DSGI:

DSGI has the ability to replay previously generated graphics output, and allows text and other graphics to be added to the output. However, DSGI does not have the ability to manage graphics output catalogs like PROC GREPLAY, so the user has to be careful as to what may exist in their graphics catalogs.

In order to insert a graphics output in DSGI, there must exist a catalog of where the graph exists, a name can also be associated with the graph, coordinate points of where to place the graphics

output, a square coordinate system, and finally the statement to draw the graph. I found that structuring code for DSGI was more complicated as compared with PROC GREPLAY at first.

When writing DSGI code there are five main steps which must be included. These five steps are as follows.³

1. initialize DSGI [GINIT();]
2. open a graphics segment [GRAPH('clear',...);]
3. generate graphics elements [GDRAW(...);]
4. close the graphics segment [GRAPH('update',...);]
5. end DSGI [GTERM();]

Once an attribute is set here, it remains until changed or code is ended. Also, once a DSGI graphics output is stored, it is possible to re-display the graphic with PROC GREPLAY.

DSGI uses a few of SAS/GRAPH's global options, such as, TITLE, FOOTNOTE, and GOPTIONS statements. With this feature, it allows DSGI to insert its own global options without creating an extra graphic output. A similar output design as seen in PROC GREPLAY is as follows. Notice DSGI does not have the ability to manage SAS catalogs.

```
FOOTNOTE1 F=SPECIAL H=2 'J J J'
          F=CENTX H=1.5 ' A LIST '
          F=WEATHER H=2 'C C C'
          F=CENTX H=1.5 ' B LIST '
          F=SPECIAL H=2 'D D D'
          F=CENTX H=1.5 ' C LIST '
          F=SPECIAL H=2 'A A A'
          F=CENTX H=1.5 ' D LIST ' ;
TITLE1 F=CENTX H=3 'DSGI ABILITIES';

DATA NULL ;
/** SET CATALOG WHERE OUTPUT EXISTS ***/
RC=GSET('CATALOG','SASDATA','P2');

RC=GINIT();
RC=GRAPH('CLEAR','SCALE');

RC=GSET('VIEWPORT', 1, .10, .10, .47,
        .48);
RC=GSET('WINDOW', 1, 10, 0, 100, 100);
RC=GSET('TRANSNO', 1);
RC=GRAPH('INSERT','PLOT3');

RC=GSET('VIEWPORT', 2, .10, .52, .47,
        .90);
RC=GSET('WINDOW', 2, 10, 0, 100, 100);
RC=GSET('TRANSNO', 2);
RC=GRAPH('INSERT','PLOT4');

RC=GSET('VIEWPORT', 3, .53, .52, .90,
        .90);
RC=GSET('WINDOW', 3, 10, 0, 100, 100);
RC=GSET('TRANSNO', 3);
RC=GRAPH('INSERT','PLOT5');

RC=GSET('VIEWPORT', 4, .53, .10, .90,
        .48);
```

```

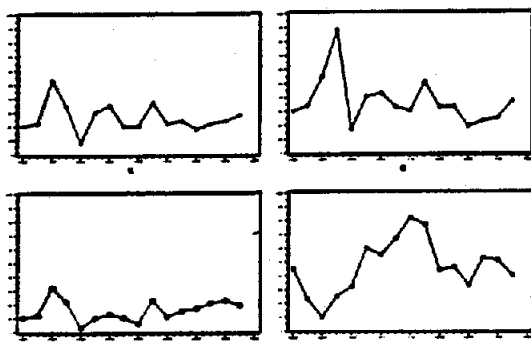
RC=GSET('WINDOW', 4, 10, 0, 100, 100);
RC=GSET('TRANSNO', 4);
RC=GRAPH('INSERT', 'PLOT6');

RC=GRAPH('UPDATE');
RC=GTERM();
RUN;

```

The output put for the above code is as follows,

DSGI ABILITIES



... ALX ... BLX ... CLX ... DLX ...

The same graphs can be generated with both graphics tools. So far both methods tend to be quite comparable, lets now take a look into the customization features of these tools.

CUSTOMIZATION AND OPTIONAL FEATURES OF PROC GREPLAY:

With PROC GREPLAY, customization is very limited, in fact, the only type of customization which could be done would be to use an ANNOTATE statement prior to running PROC GREPLAY, and at that the Annotate data can not be called by PROC GREPLAY. The procedure also allows the user to choose and design their own color maps (CMAP) and design their own Templates (TC) and that is about the extent of customization for this procedure.

When replaying graphics output, PROC GREPLAY allows the templates to overlay (as seen in an earlier example). The following is an examples of some overlaying templates and plots.

```

FOOTNOTE1 F=SPECIAL H=2 'J J J'
          F=CENTX H=1.5 ' A LIST '
          F=WEATHER H=2 'C C C'
          F=CENTX H=1.5 ' B LIST '
          F=SPECIAL H=2 'D D D'
          F=CENTX H=1.5 ' C LIST '
          F=SPECIAL H=2 'A A A'
          F=CENTX H=1.5 ' D LIST '
TITLE1 H=2.5 F=CENTX 'PROC GREPLAY
                        CUSTOMIZATION';
PROC GSLIDE GOUT=PAPER.P2 DES='TITLES FOR
GREPLAY3 PROGRAM' NAME='SLIDE1';
RUN;

```

```

PROC GREPLAY NOFS;
IGOUT=PAPER.P2;
GOUT PAPER.OVERLAY;
TC=TEMPCAT;
TC=TEMPCAT;
TC=TEMPCAT;
TC=TEMPCAT;
TDEF OVERLAY DES='HOW GREPLAY OVERLAYS
TEMPLATES FOR SUGI 17 PAPER'

```

```

1/LLX= 0 LLY= 0
  ULX= 0 ULY=100
  URX=100 URY=100
  LRX=100 LRY= 0

```

```

2/LLX= 5 LLY= 5
  ULX= 5 ULY= 95
  URX= 75 URY= 95
  LRX= 75 LRY= 5

```

```

3/LLX= 70 LLY= 75
  ULX= 70 ULY= 90
  URX= 90 URY= 90
  LRX= 90 LRY= 75

```

```

4/LLX= 70 LLY= 55
  ULX= 70 ULY= 70
  URX= 90 URY= 70
  LRX= 90 LRY= 55

```

```

5/LLX= 70 LLY= 35
  ULX= 70 ULY= 50
  URX= 90 URY= 50
  LRX= 90 LRY= 35

```

```

6/LLX= 70 LLY= 15
  ULX= 70 ULY= 30
  URX= 90 URY= 30
  LRX= 90 LRY= 15;

```

```

PREVIEW OVERLAY;
TEMPLATE OVERLAY;
TREPLAY 1:SLIDE1 2:PLOT2 3:PLOT3
        4:PLOT4 5:PLOT5 6:PLOT6;

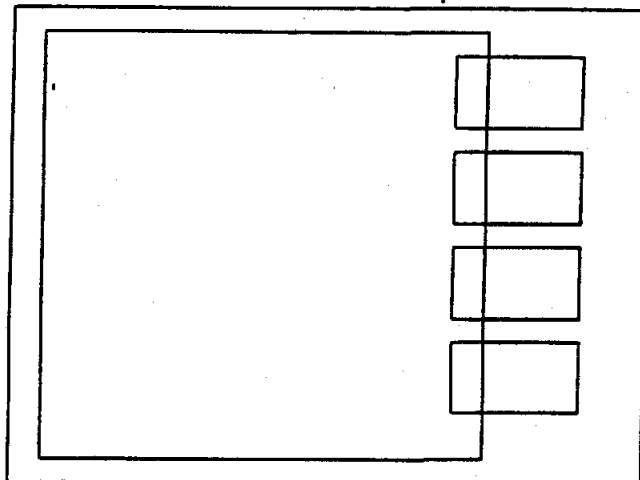
```

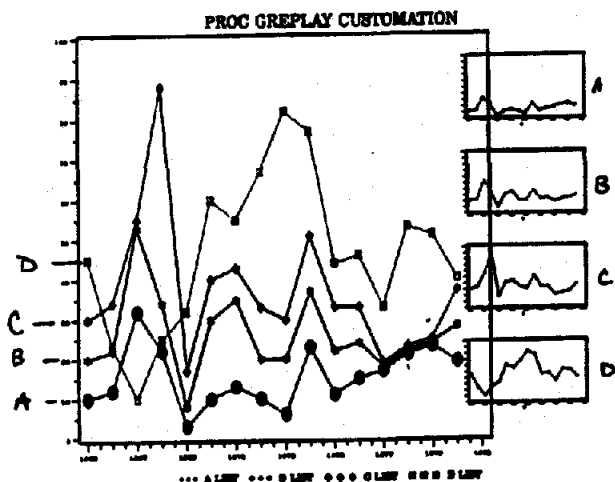
```

RUN;
QUIT;

```

The following shows both a preview of the templates and then the actual output which will be presented below and on the next page.





CUSTOMIZATION AND OPTIONAL FEATURES OF DSGI:

SAS/GRAPH's DSGI has an enormous ability for customizing graphics. This ability was shown very well in a paper presented at SUGI 14.⁵

DSGI allows the user to generate a variety of objects. The drawing elements are associated with the GDRAW function and associated attributes are listed in SAS/GRAPH's V1, p. 615. The available objects are listed as follows,

- | | |
|---------------|--|
| 1. ARCS | 6. MARKERS |
| 2. BARS | 7. PIE SLICES |
| 3. ELLIPSES | 8. POLYGONS |
| 4. ELLIP ARCS | 9. TEXT |
| 5. LINES | 10. F I L L E D
E N C L O S E D
A R E A S. |

with these objects, the user has the ability to specify color, pattern, size, style, and position of the graphics elements.⁴ The following is an example of using some of the above mentioned objects.

DATA DSGIDRAW;

```
RC=GSET('CATALOG','TEMP','PLOTS');
RC=GINIT();

RC=GRAPH('CLEAR','STAR');

RC=GSET('COLREP',1,'GREEN');
RC=GSET('COLREP',2,'BLUE');

RC=GSET('TEXCOLOR',1);
RC=GSET('TEXTFONT','XSWISS');
RC=GSET('TEXHEIGHT',5);
RC=GDRAW('TEXT',55,90,'Filled Star');

RC=GSET('TEXHEIGHT',4.5);
RC=GDRAW('TEXT',47,85,'Created with
```

DSGI');

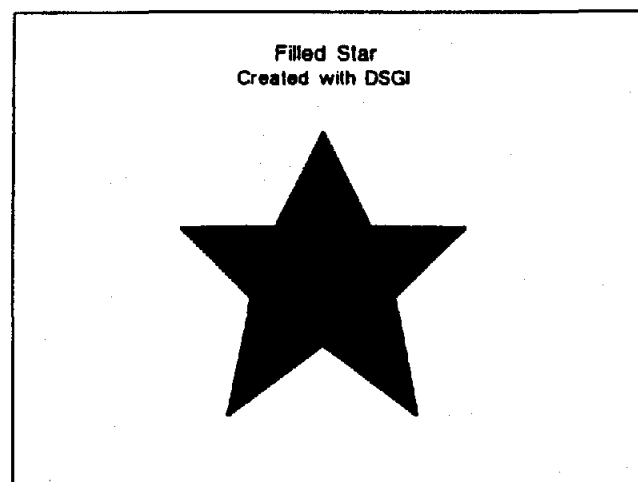
```
RC=GSET('LINCOLOR',2);
RC=GSET('FILCOLOR',1);
RC=GSET('FILTYPE','SOLID');
RC=GSET('LINWIDTH',4);
RC=GDRAW('LINE',2,35,55,55,55);
RC=GDRAW('LINE',2,55,65,55,75);
RC=GDRAW('LINE',2,75,65,55,75);
RC=GDRAW('LINE',2,75,95,55,55);
RC=GDRAW('LINE',2,80,95,40,55);
RC=GDRAW('LINE',2,85,80,15,40);
RC=GDRAW('LINE',2,85,65,15,30);
RC=GDRAW('LINE',2,45,65,15,30);
RC=GDRAW('LINE',2,45,50,15,40);
RC=GDRAW('LINE',2,50,35,40,55);
```

```
RC=GDRAW('FILL',,35,55,55,65,65,75,75,
          95,95,80,
          80,85,85,65,65,45,45,
          50,50,35,
          55,55,55,75,75,55,55,
          55,55,40,
          40,15,15,30,30,15,15,
          40,40,55);
```

```
RC=GRAPH('UPDATE');
RC=GTERM();
```

RUN;

The output generated from this code is a filled star as seen below.



DSGI has the ability to bring in a graphics output and add some character and design to meet those tough business standards of today and to make the graph stand out. The following code uniquely illustrates required output, prepared for presentation.

```
PATTERN1 V=E C=RED;
AXIS1 LABEL=(H=4 F=CENTX 'YEAR' );
AXIS2 LABEL=(A=270 R=90 F=CENTX H=4 'LIST
A')

MAJOR=(H=3 NUMBER=3)
MINOR=(H=1.5 NUMBER=3)
ORDER=0 TO 40 BY 5;

PROC GCHART DATA=PAPER.GRAPH
GOUT=PAPER.P2;
VBAR YEAR /DISCRETE SUMVAR=A
```

```

DES=' CHART CREATED FOR DSGI
TO DO CUSTOMIZING'
WIDTH=3 MAXIS=AXIS1
RAXIS=AXIS2
NAME='CHART1';

RUN;

TITLE1 F=CENTX H=3 'DSGI CUSTOMIZING
FEATURES';
TITLE2 F=CENTX H=2 'LIST A';

DATA _NULL_;

/**** SET CATALOG WHERE OUTPUT EXISTS ****/

RC=GSET('CATALOG','PAPER','P2');

RC=GINIT();
RC=GRAPH('CLEAR','CHART1','Customizing
DSGI charts');
RC=GSET('COLREP',1,'RED');
RC=GSET('COLREP',2,'BLACK');

RC=GSET('FILCOLOR',1);
RC=GSET('FILTYPE','HOLLOW');
RC=GDRAW('PIE',15,40,2,0,360);
RC=GDRAW('PIE',22.5,38,2,0,360);
RC=GSET('FILCOLOR',2);
RC=GSET('FILTYPE','SOLID');
RC=GDRAW('PIE',30,52,2,0,360);
RC=GDRAW('PIE',37.5,42,2,0,360);
RC=GSET('FILCOLOR',1);
RC=GSET('FILTYPE','HOLLOW');
RC=GDRAW('PIE',45,23,2,0,360);
RC=GDRAW('PIE',52,30,2,0,360);
RC=GDRAW('PIE',59.5,52,2,0,360);
RC=GDRAW('PIE',67,30,2,0,360);
RC=GDRAW('PIE',74.5,26,2,0,360);
RC=GSET('FILCOLOR',2);
RC=GSET('FILTYPE','SOLID');
RC=GDRAW('PIE',82,43,2,0,360);
RC=GSET('FILCOLOR',1);
RC=GSET('FILTYPE','HOLLOW');
RC=GDRAW('PIE',89,35,2,0,360);
RC=GSET('FILCOLOR',2);
RC=GSET('FILTYPE','SOLID');
RC=GDRAW('PIE',96.5,35,2,0,360);
RC=GDRAW('PIE',104,38,2,0,360);
RC=GDRAW('PIE',111.5,44,2,0,360);
RC=GDRAW('PIE',119,49,2,0,360);
RC=GSET('FILCOLOR',1);
RC=GSET('FILTYPE','HOLLOW');
RC=GDRAW('PIE',126,55,2,0,360);

RC=GDRAW('PIE',111,85,2,0,360);
RC=GSET('TEXCOLOR',2);
RC=GSET('TEXTFONT','CENTX');
RC=GSET('TEXHEIGHT',2);
RC=GDRAW('TEXT',116,84,'QUOTA NOT
MET');

RC=GSET('FILCOLOR',2);
RC=GSET('FILTYPE','SOLID');
RC=GDRAW('PIE',111,91,2,0,360);
RC=GSET('TEXCOLOR',2);
RC=GSET('TEXTFONT','CENTX');
RC=GSET('TEXHEIGHT',2);
RC=GDRAW('TEXT',116,90,'QUOTA MET');

RC=GSET('VIEWPORT',1,.15,.15,.90,.80);
RC=GSET('WINDOW',1,30,20,85,83);
RC=GSET('TRANSNO',1);
RC=GRAPH('INSERT','CHART1');

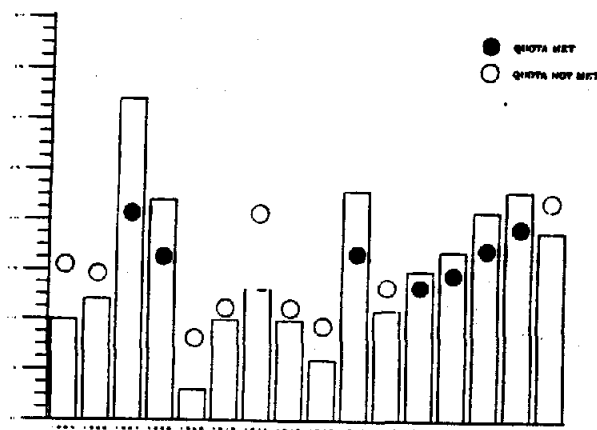
```

```

RC=GRAPH('UPDATE');
RC=GTERM();
RUN;

```

DSGI CUSTOMIZING FEATURES
LIST A



CONCLUSION:

This paper has compared two methods available in SAS/GRAPH's which have the ability to replay graphs. For PROC GREPLAY the two major distinguishing features would be,

- the windowing feature
- the ability to manage graphic catalogs.

DSGI has enormous capabilities, and the two major features would be,

- the ability to zoom in on a particular part of a graph
- the ability to customize a graph generated by SAS.

These features make the products unique and depending on your needs, will depend on what process is used.

Trademark Notice,

* SAS is a registered trademark of SAS Institute Inc., Cary, NC, USA.

References,

- SAS/GRAPH, V2, p 1192
- SAS/GRAPH, V1, p 600
- SAS/GRAPH, V1, p 604
- SAS/GRAPH, V1, p 603
- SUGI 14 Proceedings, "The DATA Step Graphics Interface", Jade Walker, pp 561-556.

Author,

Lynn M Thome-Polingo
G.D. Searle
4901 Searle Parkway
Skokie, IL 60077
(708) 982-4872