A MACRO-DRIVEN SYSTEM FOR BACKGROUND GRAPH GENERATION

Robert W. Phelps
NCBH - BGSM

Abstract:

The Executive Financial DataBase of Bowman Gray School of Medicine of Wake Forest University has evolved into a system containing over 7,000 entities, with over 5,000 graphs produced yearly. To aid in the development of the system, SAS/AF®, SAS/FSP®, SAS Graph® and the SAS Macro Language® were used. A process for generating graphs in the background was added to aid in the work.

Introduction:

The Executive Financial DataBase of Bowman Gray School of Medicine of Wake Forest University has evolved into a system containing over 7,000 entities with annual data elements dating from 1954. Over 5,000 graphs are produced each year, which provide strategic information on funding sources and usage, research and development, salary and budget.

Originally written in interactive Fortran, the system was implemented using SAS Software in 1987. Several enhancements were added, including a background process for the generation of graphs.

Graph Generation:

SAS/AF, SAS/FSP, SAS Macro Language and SAS/Graph Software were used in the development of the system. A series of customized menus and panels prompt the user for parameter data (variable name, title, line labels, graph colors, etc.) needed to produce a graph. The process is repeated annually with many of the same graphs generated with only the year ranges changed. Because the generation of several thousands of graphs interactively is too labor-intensive, a background "batch" process was developed using the following guidelines:

1. Keep the user interface consistent.
2. Provide a mechanism to track the graphs that are needed without having to re-key the graph parameters.
3. Preserve the use of the code already written to make maintenance easier and development more timely.

A "batch" is created using SAS/FSP with a panel (see Figure 1) designed to resemble the SAS/AF screens used for the online generation of graphs. After creation, one or more "batches" may be submitted to generate the graphs. The submission process concatenates the individual batches into a master batch which acts as a work queue from which subsequent jobs process. The code that runs the batch process can be seen in Figure 2. An embedded TSO command submits the JCL to the system. The macros used are explained in the next section. After the submitted jobs have completed, the graphs may be viewed or sent to a hardcopy device by selecting an option that invokes Proc GREPLAY and by entering the batch name as the input graphic catalog name. A typical graph may be seen in Figure 3.

Processing Macros:

%INIT - Sets up the environment for the batch run. Global variables are initialized and formats created.

%PUTSYM - (Fig 4) Data step access with the point option is used to read observations in the "batch" dataset one at a time. Several CALL SYMPUT statements output the values of the macro variables used in later steps.

%EXTRACT - Obtains the data needed for the selected year range.

%PREPARE - Processes the data pulled by %EXTRACT and calculates the maximum and minimum values, etc. to be used in subsequent steps to format and scale the graph.

%ANNOTATE - Determines the placement of the line labels and outputs the annotate data set used by the GPLOT procedure.

%PLOT - Generates the plot using the GPLOT procedure.

Conclusions:

The use of SAS/AF, SAS/FSP, and SAS/Graph, and the Macro Language add system design flexibility.

The addition of background "batch" processing makes the system capable of high volume output of graphic information.

The use of SAS/FSP in storing parameter data for later use in a batch process can be carried over to any function requiring reoccurring input.

For More Information:

Robert W. Phelps
Decision Support Systems
NCBH - BGSM
300 S. Hawthorne Road
Winston-Salem, NC 27103
(919) 748-4760

SAS, SAS/AF, SAS/FSP AND SAS/Graph are registered trademarks of SAS Institute Inc., Cary, NC, USA.
### AF panel

**TITLES AND LABELS**

- **Title**: DEPARTMENTAL EXPENSE
- **Sub-title**: TIME (IN YEARS)
- **Footnote**: THOUSANDS OF DOLLARS
- **X-Label**: TIME (IN YEARS)
- **Y-Label**: THOUSANDS OF DOLLARS

**X Description**

<table>
<thead>
<tr>
<th>Y</th>
<th>File</th>
<th>Variable</th>
<th>Color</th>
<th>Label</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DEPT-X</td>
<td>TOPEXP</td>
<td>WHITE</td>
<td>TOTAL</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DEPT-X</td>
<td>SALEXP</td>
<td>RED</td>
<td>SALARIES</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DEPT-X</td>
<td>SUPEXP</td>
<td>GREEN</td>
<td>SUPPLIES</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>BLUE</td>
<td>EQUIPMENT</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>RED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>GREEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>BLUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>WHITE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>GREEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Special features**

- **PF/3** to submit
- **PF/12** to cancel
- Print values to Screen or File or Send directly to Printer
- Percent of total
- Compute the total X

**Print values to Screen or File or Send directly to Printer**

- Press PF/B for more

### FSP panel

**Edit SAS data set: BAT.SUGI**

**Command ==>**

**Screen 1**

<table>
<thead>
<tr>
<th>Y</th>
<th>File</th>
<th>Variable</th>
<th>Color</th>
<th>Label</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DEPT-X</td>
<td>TOPEXP</td>
<td>WHITE</td>
<td>TOTAL</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DEPT-X</td>
<td>SALEXP</td>
<td>RED</td>
<td>SALARIES</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DEPT-X</td>
<td>SUPEXP</td>
<td>GREEN</td>
<td>SUPPLIES</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>BLUE</td>
<td>EQUIPMENT</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>RED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>GREEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>BLUE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>WHITE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td>GREEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DEPT-X</td>
<td>EQEXP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Obs 1**

PF1=Help, PF3=Save/End, PF6=Dup, PF7=Previous, PF8=Next, PF9=Add

PF12=Cancel

**Figure 1**
/* THIS PROGRAM RUNS THE BACKGROUND PROCESS FOR EACH "BATCH"
/* IT IS SUBMITTED ALONG WITH THE NECESSARY JCL VIA AN AF PANEL OPTION */

%INIT;  /*SET UP GLOBAL VARIABLES NEEDED FOR EXECUTION*/
/* DATA STEP TO GET A BATCH FROM THE QUE */
/* SUBSEQUENT SUBMITTED JOBS WILL PROCESS THE REMAINING BATCHES*/

DATA BAT.BATCH(DROP=HLDBAT)
  WORK.BATCH(DROP=HLDBAT);
SET BAT.BATCH;
RETAIN HLDBAT;
IF _N_ = 1 THEN
  DO;
    HLDBAT=NAMBAT;
  END;
IF NAMBAT=HLDBAT THEN
  OUTPUT WORK.BATCH;
ELSE
  OUTPUT BAT.BATCH;
/* DATA STEP TO OUTPUT MACRO VARIABLES CONTAINING THE NUMBER OF */
/* OBSERVATIONS IN THE BATCH AND THE NAME OF THE GRAPHIC CATALOG */
/* TO STORE THE GRAPHS PRODUCED BY THIS BATCH. */

DATA WORK.BATCH;
SET WORK.BATCH END=FINI;
IF FINI THEN
  DO;
    CALL SYMPUT('BATCNT', LEFT(PUT(_N_,3.)));  
    CALL SYMPUT('PLOTCAT', BAT.||NAMBAT);
  END;
RUN;
/* GENERATE THE MAIN MACRO TO PERFORM THE PROCESSING*/
%MACRO RUNIT;  
  %DO I = 1 %TO &BATCNT;  
  %PUTSYM(&I);  
  %EXTRACT;  
  %IF &NREC >0 %THEN %DO; /*PERFORM PLOT ONLY IF RECORDS FOUND*/  
  %PREPARE;  
  %ANNOTATE;  
  %PLOT;  
  %END;  
%END;  
%MEND RUNIT;  
%RUNIT;  /*RUN THE GENERATED MACRO*/

Figure 2

664
DEPARTMENTAL EXPENSE

Figure 3
%MACRO Putsym(obs);
%local obs;
%* this macro uses symput to output the global variables needed to;
%* produce a line graph;

data _nu11_
array files(10) $ f11-f110;
run= &obs;
set bat. batch point=n;
if _error_ then abort;
call symput('titl',trim(titl));
call symput('tit2',trim(tit2));
call symput('titlc',titlc);
call symput('foot',trim(footc));
call symput(xlab,trim(xlab));
call symput('yc',yc);
call symput('print',print);
call symput('xmin',left(xmin));
call symput('yc',yc);
call symput('yc',yc);
call symput('yc',yc);
call symput('yc',yc);
call symput('yc',yc);
call symput('yc',yc);
call symput('yc',yc);
call symput( 'fun2',fun2);
call symput( 'fun4',fun4);
call symput( 'fun6',fun6);
call symput( 'fun8',fun8);
call symput( 'fun10',fun10)
/* count the number of lines to graph */
LINES=0;
I = 1;
DO UNTIL(I > 10);
IF FILES(I) = ' ' THEN
   DO;
      I = I - 1; /* first blank line indicates end of input*/
      call symput('LINES',LEFT(I));
   I = 10;
   END;
   I = I + 1;
END;
stop;
rnn;
%MEND Putsym;

Figure 4