Creating High Resolution Graphics Interactively
Using BASE/SAS® SAS/ACCESS® IDMSExt SAS/AF® and SAS/GRAPH®
Matthew D. Leet
General Electric Silicone Products
Division of GE Plastics
Waterford N.Y. 12188

Abstract

This article will demonstrate a system for producing high resolution graphics interactively using SAS/ACCESS, SAS/GRAPH®, and SAS/AF products. A discussion of IDMS/R Logical Record Facility (LRF) and CLISTS will also be included.

The recent release of SAS/ACCESS IDMSExt procedure which incorporates the 'where clause' feature, available through the IDMS/R LRF, enables the SAS programmer to access the IDMS database from within a SAS batch job or on-line interactively. The benefits of this new release are that applications which previously had to run off batch COBOL extract files can now be made interactive saving computer resources and improving the quality of the data for the user. I will attempt through the use of the following example to outline the steps required in order to develop an interactive application using SAS/AF, SAS/ACCESS, BASE SAS and SAS/GRAPH.

Defining the Logical Record for the Usage Total Database.

Usage total records are used to keep track of a variety of actual and estimated business related activities. In an MRP system forecasts are used in conjunction with a number of other consumption and production data to develop a Master Production Schedule which the planners use to help maintain manufacturing schedules. It is desirable to be able to analyze historical data concerning the accuracy of the forecast to help revise/adjust the forecast information in order to optimize the schedule. An overlay plot is an excellent tool by which a manager can quickly review the accuracy and spot trends in demand for a product. To illustrate the use of the IDMSExt procedure I chose to produce a Gplot of forecasted versus actual orders for a product and year of interest. The database records required to obtain the data are as follows:

Using the on-line subschema compiler (SSC from ENTe) the following code was used for creating an IDMS/R logical record. For more information on creating logical records using LRf refer to Cullinet's Logical Record Facility Database Administrators Guide.

```plaintext
PART-MASTER CALC
PART-NUMBER-001
CAS-BOM-AREA

PART-PLANT NPO MA
ASC PLANT-CODE-010

PLANT-LOCATION
VIA
CAS-BOM-AREA

PLANT-USAGE
NPO MA FIRST

USAGE-TOTALS
VIA
CAS-INV-AREA

USING THE ON-LINE SUBSCHEMA COMPILER (SSC FROM ENTe) THE FOLLOWING CODE WAS USED FOR CREATING AN IDMS/R LOGICAL RECORD. FOR MORE INFORMATION ON CREATING LOGICAL RECORDS USING LRf REFER TO CULLINET'S LOGICAL RECORD FACILITY DATABASE ADMINISTRATORS GUIDE.
```
With the logical record created the database could now be accessed through the LRF using SAS IDMSEXTP procedure.

Creating the MAPIN dataset

In order to allow the SAS/AF program a way to request usage total records for a specific product it was necessary to create a MAPIN dataset to the IDMSEXTP procedure. This dataset is used to provide a model 'where clause' which gets modified by the on-line request program to subsequently CALC into the IDMS/R database. The MAPIN dataset was created by invoking proc IDMSEXTP from the SAS Display Manager. In order to store the MAPIN dataset permanently a SAS Library dataset must be allocated to your session to save the output of the IDMSEXTP procedure. The following command can be used to do the allocation of a permanent SAS library directly from the ISPF option 6 command line:

ALLOC DSNAME('your.dataset') NEW CAT SPACE (P.S) TRACKS 8K SPACE (23760) DSORG (DA) REC FM (U) LRECL (32765) UNIT (SYSDA)

P=primary allocation S=secondary allocation

Note - some options may vary for your site

For more information on allocating datasets refer to IBM's MVS/Extended Architecture ISPF Command Language Reference. For more information on SAS Libraries refer to the SAS Users Guide Basics. It was also necessary to allocate the central version and database files associated with the IDMS logical record and Schema of interest. The following are TSO commands issued from SAS to allocate the development manufacturing database and central version:

TSO ALLOC F(SYSTTL) DSN='your data file name') SHR REU: RUN:
TSO ALLOC F/DICTDBD OS=FDS 'your dictionary file name') SHR REU: RUN:

Then to invoke proc IDMSEXTP:

PR0C IDMSEXTP; RUN;

The IDMS/R DATA ACCESS PANEL is displayed (screen 1):

The SAVE SELECTIONS option must be changed to 'Y' to create the MAPIN dataset and the OUTPUT SAS DATA SET is of the form libref.name where libref refers to the name of the SAS library in the allocate statement and name is up to 8 characters of choice (National characters non-numeric first char). Enter the name of the LRF subschema, schema name, LRF record name, version no and dictionary and press enter. The IDMS/R selection screen will then be displayed (screen 2):

```
library name: libref name
data set name: MAPIN
```

Select those elements you wish to include in the MAPIN dataset and supply the SAS variable name and format desired. If you do not supply a SAS name the procedure uses the first 8 characters of the IDMS variable name. If you wish to change the format you can type over the one supplied by the IDMSEXTP procedure and that format will be the one used to do the extraction. For more information on formats refer to the SAS Users Guide Basics. Once the elements are selected and the SAS names and formats provided the WHERE CRITERIA can be specified. Where criteria will support the following arithmetic expressions: +, -, *, /, (and may use parenthesis to change the order of evaluation. Do not include the word 'WHERE' however in the where clause. Boolean operators (AND, OR, NOT) are permitted and KEYWORDS can be supplied if defined by the subschema[]. After entering the where criteria pressing PF3 will cause the MAPIN dataset to be created and saved. You can view the contents of this dataset if you wish by issuing a PROC PRINT as follow:

PROC PRINT DATA=libref.name; RUN;

(press PF3)
Had not the SAVE SELECTIONS option been chosen IDMSEXT would have performed the extraction from the database and created a SAS dataset with the selected variables. That dataset could have been used as input to subsequent SAS data steps and procs. But right now the goal was to create the MAPIN dataset to be used later on.

Using SAS/AF to build the request/display program

Once the MAPIN dataset was created the next step was to build the request/display program. Again a permanent SAS library is needed to store the SAS program in, otherwise it will be deleted when you exit SAS. The PROC BUILD function is used to create a SAS program. To invoke simply type the following from the program editor (bottom half) of the SAS Display Manager session.

PROC BUILD CATALOG=libref.catalog: RUN;

Where libref is the dname of the permanent SAS library to store the SAS program and catalog is the name of a catalog to store programs in. The directory screen is then displayed.

To create the SAS program in the catalog type the following on the command line of the directory screen:

EDIT name.type (press enter)

Where name is the program name and type is PROGRAM for type program.

The following edit screen is displayed:

```
Type NUMS ON on the command line and line numbers will be added to the left hand side of the screen. Most TSO line mode commands will function as they do in TSO. For a complete list of editing commands refer to the SAS/AF Users Guide.

The following is the source code used to perform the extract using the product and year entered into the SAS/AF request screen.

The SAS macro facility was used to substitute the part number to modify the MAPIN dataset 'where clause' so the MACRO option must be in effect in order for the symbolic substitution to take place. Macro variables begin with the 'i' character. In order to have the MACRO facility in effect during development and debugging use the following SAS invocation command from the TSO command facility:

SAS OPTIONS ('MACRO MACROGEN SYMBOLGEN')
```
The MACROGEN and SYMBOLGEN options are for debugging purposes so that you can see the substitution by the MACRO facility in the SAS log after execution. The SAS log is the top half of the display manager session and is where SAS logs all of the program editor statements that are submitted to SAS as well as any errors returned from compilation or execution.

Appendix A is a PROC PRINT of the raw data as it was extracted by the LRF and put into the EXTFILL SAS dataset. Following the modification of the MAPIN dataset and the IBMSEXT PROC are two DATA steps. The first(EXITFL2) selects the records to be processed while the second splits each month out into a separate workfile by month. PROC's sort and merge are used to join the 2 workfiles by month so that GLOT can graph the data. The LEGEND data step is used to create an Annotate dataset for drawing the legend on the graph where I wanted it to appear. For more information on the GLOT procedure and Annotate datasets refer to the SAS/GRAPH Users Guide.

CLIST Driver

Once all of the pieces were in place I created a CLIST (command list) to allocate the datasets and invoke SAS. The CLIST passes as a parameter the DISPLAY command which SAS/AF recognizes and executes immediately. The display command displays a menu which was built using PROC BUILD in a similar manner to the SAS program. The type however was MENU rather than program. For more information on menu's refer to the SAS/AF User's Guide. For more information on CLIST's refer to IBM's TSO Extensions CLIST'S: Implementation and reference.
Summary

BASE/SAS, SAS/ACCESS, SAS/GRAPH, and SAS/AF combined with the IDMS/R LRF 'where clause' processing provide a powerful set of tools to analyze important business data in a real time on-line environment. The new release of SAS ACCESS which incorporates IDMS/R 'where clause' processing enables the SAS programmer to analyze data interactively. This improves the quality of the data and eliminates the need for batch extracts to obtain information from the database. As a result SAS products are becoming one of the primary end user tools at GE Silicone Products. We currently are developing a front end to the SAS IDMSEXT using SAS/AF which will make the process of extracting data from the IDMS/R totally transparent to the user. It will perform the operations to prefill the IDMSEXT data access panel based on the users selection from a menu of database choices. Each of these choices will have an IDMS logical record defined as well as a predefined MAPIN dataset. This way a non SAS/IDMS user will be able access data that resides in the IDMS/R database. Once data has been extracted it will be available for further analysis through a SAS/AF menu system which will build the program code needed to do the analysis.

References


SAS, SAS/AF, SAS/ACCESS and SAS/GRAPH are registered trademarks of SAS Institute Inc., Cary, NC, USA. IDMS/R is a registered trademark of Cullinet Software Inc., Westwood Mass., USA

Contact

Matthew D. Leet
General Electric Silicone Products
Division of GE Plastics
Hudson River Road
Waterford, N.Y. 12188
(518) 233-2566