Paper 122X

A Non-Standard Report Card - Informing parents about what their children know.

Daniel Ralyea, Rock Hill Schools

ABSTRACT

A standards based report card represents a non-traditional use of the power of SAS. A typical report card lists the subject and a letter or a number grade. It does not identify the skills that lead to that grade. SAS allows us to read the grades from PowerSchool's Oracle database combine in test scores from an outside vendor and summarize multiple grades into more generalized standards. Using SAS ODS individual report cards are sorted by school and teacher and printed for each student. Developing this report card poses several challenges: producing grade level specific graphics, developing and distributing the reports.

INTRODUCTION

This paper will explain how to produce a grade level specific graphic using proc ganno, how to develop a macro that will create a pdf document for each school organized by grade level and homeroom where each student's report card is printed on one double sided page.

PREPARING THE GRAPHICS

Proc ganno produces a graphic based on a dataset. The dataset controls the position of the "pen" and contains the instructions on how to draw the graphic.

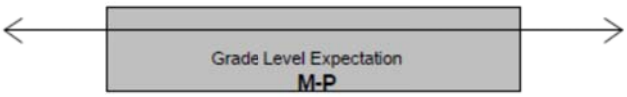

Figure 1. Grade Level Expectation Graphic

This is a simple graphic composed of a filled rectangle, a label for the text "Grade Level Expectation", a second label that changed by grade level indicating the levels for that grade (in this case M-P) and five lines to form the double arrow.

The drawing is done by declaring the graphics options. These options set up the drawing pallet.

```
goptions hsize =5in vsize=1in vpos=12 hpos=42;
```

This declares an area 5 inches wide 1 inch deep that can be thought of as a 42X12 grid with (0,0) in the lower left corner. The position of the pen is then dictated by the x and y values in the dataset. Creating the dataset was done in two phases. First the static elements were defined in one dataset. The variables Function, X, Y, HSYS, VSYS, STYLE, COLOR, POSITION, SIZE, LINE, and TEXT are created using a format statement. The components of the graphic are then created by using the function and associated attributes to move the pen. Function= "POLY" starts a polygon. The x and y coordinates determine the starting position. The color attribute refers to the fill color of the polygon and the style is the fill pattern. The next line Function="POLYCONT" sets the next vertex and the color of the border. The next two points are declared and the final side is completed automatically.

The next section of the dataset is the placement of the text. Function="LABEL" declares the placement of a label. The position value determines the placement relative to the x and y values. Position=6 aligns the text centered vertically to the left of the point.

The arrow is composed of five line segments. Each segment is created by a pair of commands. First move the pen to the starting point the draw the segment. The draw function then connects the starting point from the move function to the ending point indicated in the draw row.

```
Data LiteracyBOX;
/*Declare Variables*/
```

```
Format Function $8. X 3. Y 3. HSYS $1. VSYS $1. STYLE $10.  COLOR $10.
POSITION $1. SIZE 6.2   LINE 1. TEXT $150.;
HSYS=6;     VSYS=6;
/*Create Rectangle*/
Function="POLY";         X=10; Y=1;  COLOR="LIGR"; STYLE="MSOLID"; OUTPUT;
Function="POLYCONT";     X=10; Y=8; COLOR="BLACK";     OUTPUT;
FUNCTION="POLYCONT";     X=34; Y=8;  OUTPUT;
FUNCTION="POLYCONT";     X=34; Y=1;  OUTPUT;
/*Create Label*/
POSITION="6"; STYLE="ALBANY AMT";
FUNCTION="LABEL";  X=16;      Y=4; TEXT="Grade Level Expectation"; size=1.5;
OUTPUT;
/*Create Arrow*/
Function="MOVE";     X=4;      Y=6;  OUTPUT;
Function="DRAW";     X=40;     Y=6;  OUTPUT;
Function="MOVE";     X=4;      Y=6;  OUTPUT;
Function="DRAW";  X=5;  Y=7;  OUTPUT;
Function="MOVE";  X=4;  Y=6;  OUTPUT;
Function="DRAW";  X=5;  Y=5;  OUTPUT;
Function="MOVE";  X=40; Y=6;  OUTPUT;
Function="DRAW";  X=39; Y=7;  OUTPUT;
Function="MOVE";  X=40; Y=6;  OUTPUT;
Function="DRAW";  X=39; Y=5;  OUTPUT;
Run;
```

This creates the static portion of the dataset. In order to make the graphic sensitive to the grade level being reported on the dataset needs to be modified. The text listing the level of the expectation changes by grade level and reporting period. We will create a macro that modifies the dataset based on input parameters.

The macro literacy accepts two parameters: gradelow and gradehigh. These are the letters that reflect the expected reading levels for the students at that point in time. The dataset templev consists of a single observation that is a label dictated by `TEXT=compress("&gradelow"||"-"||"&gradehigh");`

The static dataset templev is appended to literacybox forming the dataset literacy. The final step in the macro runs proc ganno which executes the steps listed in the dataset literacy.

Figure 1 is the result of the command `%Literacy("M", "P");`.

```
%macro Literacy(gradelow,gradehigh);
      data templev;
      Format Function $8. X 3. Y 3. HSYS $1. VSYS $1. STYLE $15.  COLOR $10.
      POSITION $1. SIZE 6.2   LINE 1. TEXT $150.;
      HSYS=6;     VSYS=6; STYLE="ME";COLOR="BLACK";
      POSITION="6"; STYLE="ALBANY AMT/BOLD";
      FUNCTION="LABEL"; X=21; Y=2; TEXT=compress("&gradelow"||"-
      "||"&gradehigh"); size=1.8;OUTPUT;

      data literacy;
      Format Function $8. X 3. Y 3. HSYS $1. VSYS $1. STYLE $15.  COLOR $10.
      POSITION $1. SIZE 6.2   LINE 1. TEXT $150.;
      Set literacybox templev;
      Run;

      proc ganno anno=literacy; run;
%mend;
```

Because the program is designed to produce selected report cards as well as running district wide report cards the parameters are set inside the following data statement. &grade is pulled from the demographic portion of the student information and the parameters are set for the grade level.

```
data _null_;
%if "&grade." eq "0" %then %do;
call symput("gradel","Pre");
call symput("gradeh","D");
%end;

%if "&grade." eq "1" %then %do;
call symput("gradel","D");
call symput("gradeh","J");
%end;
%if "&grade." eq "2" %then %do;
call symput("gradel","J");
call symput("gradeh","M");
%end;
%if "&grade." eq "3" %then %do;
call symput("gradel","M");
call symput("gradeh","P");
%end;

%if "&grade." eq "4" %then %do;
call symput("gradel","P");
call symput("gradeh","S");
%end;

%if "&grade." eq "5" %then %do;
call symput("gradel","S");
call symput("gradeh","V");
%end;
run;
```

So inside the program the command to produce the graphic is `%Literacy(&gradel., &gradeh.);`.

PREPARING FOR DISTRIBUTION

The task of generating report cards could produce a file for every student or a single file for the whole district both of those options can create logistical problems for users. The optimal solution for our district is to produce one file per homeroom inside a school level folder. SAS ODS will create new files but doesn't allow output to nonexistent directories. Creating those folders and generating the file names are the next step in the distribution process.

The student demographic dataset (work.students) contains the number of the school that the student attends. The following code creates a set of macro variables which contain the school number of the schools referenced in the student dataset. The code as written could create 99 macro variables but the execution will only generate the number of variables needed.

```
proc sql;
Create table fileset as Select compress(put(school_number,$2.)||home_room) as
fileid, school_number, home_room
From work.students;
quit;
proc sort noduprec;
by fileid;
proc sql;
Select distinct school_number into :school1-:school99
From work.file;
quit;
```

Whenever proc sql is run a macro variable is created containing the number of observations. The following line captures the number of observations in a macro variable Snumobs.

```
%let Snumobs=&sqlobs.;
```

The following macros create a series of folders inside a destination directory to place the report cards into then runs the code to produce the report cards in those folders. The names of the folders correspond to the school numbers retrieved from the student demographic files. Proc SQL is used again to create macro variables containing the file name and home_room. The macro variable containing the file name will be used to name the ODS destination. The home_room will be used to subset the dataset containing the student information.

```
%macro RunReport(schlnum);
proc sql;
select fileid, home_room into :filename1-:filename999,:HR1-:HR999
from fileset
where school_number eq &schlnum.;
quit;
%let Tnumobs = &sqlobs.;
%do i=1 %to &Tnumobs.;
ods pdf File=
"\\server\drive\report_cards\&Schlnum.\&&filename&i.._Report.pdf”;
.
.
.
Report code sub-setted by using where statements and the macro variable
&&HR&i..
.
.
.
%end;
%mend;
```

```
options noxwait;
%macro MakeDest();
%do i=1 %to &Snumobs.;
x md "\\server\drive\report_cards\&&School&i..”;
%RunReport(&&School&i..);
%end;
%mend;
%MakeDest();
```

The second macro calls the first macro passing the parameter containing the school number to the first macro after creating the destination directory. The RunReport macro creates temporary macro variables containing the file name and home room teacher. The macro variables only exist for that particular instance of the macro and are deleted from memory when that instance closes. The home room teacher and filename were both read from the same line of the fileset table and guarantee that all of the students in that homeroom will have be placed in that pdf file.

CONCLUSION

There were several obstacles in creating the standards based report card. They ranged from customized graphics to rigidly formatted reports placed in a pdf designed to print multi-column report on two sides of a paper. The actual report card consisted of seven distinct uses of proc report, a custom graphic applied for eight thousand students distributed across more than four hundred teachers in seventeen elementary schools.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Daniel Ralyea
Rock Hill Schools
660 N Anderson Rd
Rock Hill, SC 29730
803-981-1050
dralyea@rhmail.org