A Dashboard Application Developed Using SAS® GRAPH Proc Annotate
Timothy A. Kenney, Kenney IS Consulting, Inc., Ojai, CA

ABSTRACT
For the web-based 2005 National Healthcare Quality Report State Resources, we developed a simple speedometer application using proc Annotate and the macro %CYARROW (SUGI 30, %CYARROW, A NEW ANNOTATE MACRO” Chung and Huang) to graphically demonstrate “the State’s performance on summary measures of the quality of types of care, settings of care, and overall quality of health care relative to the region or Nation on each summary measure.” (http://www.qualitytools.ahrq.gov/qualityreport/2005/state/summary/intro.aspx). This paper discusses the process we used to narrow our development tools down to SAS®/GRAPH and PROC ANNOTATE, the steps required to move from data to graphics using standard tools available in Microsoft Windows XP PRO and SAS®, the development steps associated with shaping a final graphic, and the issues we faced in publishing the results in a web based format. The final process used the MACRO language, the SAS®/GRAPH ANNOTATE procedure, and the BASE product. Readers should have some familiarity with these tools. This work was performed under a subcontract with Thomson Medstat as part of the Healthcare Cost and Utilization Project sponsored by Agency for Healthcare Research and Quality.

INTRODUCTION
A combination of strict publishing and accessibility requirements for federally-funded web sites (http://www.access-board.gov/sec508/guide/1194.22.htm), a paucity of simple commercial solutions, and the need for mass production capability led us to developing a simple “dashboard” graphic application using Proc Annotate in SAS® Graph.

Below I will outline our application requirements, the tools we selected to implement the graphics, and the steps we took to move the process into a production stream.

MY BACKGROUND
I have been programming in BASE SAS and the Macro language for over 25 years, with brief excursions into the graphics package, as needed. The solution you will see explained below reflects my experience in developing production systems, and is not meant to be the final word in an elegant graphical approach to the problem. The reader may realize a more efficient approach, and if so, I welcome comments. This application is worth presenting for its relative simplicity and its applicability to a rigid production requirement producing a massive amount of graphical output.

APPLICATION
Our client, Agency for Healthcare Research and Quality (AHRQ) Center for Quality Improvement and Patient Safety, asked for a graphic component to a public information web site describing relative performance, by state, on a variety of health service performance measures included in the 2005 National Healthcare Quality Report (NHQR). Quality measures were summarized by types of care, settings of care, and overall quality of health care relative to national and regional benchmarks and expressed as percentages. The public can browse the NHQR State Resources web site (http://www.qualitytools.ahrq.gov/qualityreport/2005/state/summary/intro.aspx) and access the comparison information as graphs, descriptors of the graphs, and as data tables, allowing for a wide variety of accessibility and preferences. The reader is encouraged to browse this site and reflect on the disparity and uniformity of health care services in this country.

Below is a speedometer application for the state of California:

What Is California’s Overall Health Care Quality Performance Compared to All States and How Has it Changed?

The State's performance across all NHQR Quality Measures (up to 99) is shown below compared to all States in the most recent data year (solid line) and in a preceding data year (dashed line).
An arrow pointing to "very weak" means all or nearly all included measures for a State are worse than average within a given data year. Conversely, an arrow pointing to "very strong" indicates that all or nearly all available measures for a State are better than average within a given data year. The other categories scale from weak to strong performance and represent the State’s balance of worse than average, average, and better than average measures. To examine all the measures behind this performance, click on the meter. For more information on how these measures are translated into a performance meter, select Methods, or to view additional information about this State, make a selection from the menu on the left.

RESTRICTIONS
Because our target graphics needed to be incorporated into a web site, we were given the following restrictions:

- Minimize the size of each graphic file so the overall memory footprint for the Web page can stay under 12KB
- Use GIF or JPG format because these formats are easily viewed by all browsers
- Set the maximum width of the graphics to 400 pixels.

Also, we needed to produce 918 dashboard graphics (51 (50 states and D.C.) x 2 comparison groups (region and nation) x 9 summary measures) from a single summary data base incorporating the statistics required. We also understood that the measures were still under development and the dashboards would be run multiple times as the methodology for the summary measures was finalized.

WHY SAS®?
When I was ask to evaluate this project I looked at commercially available products from SAS® and other commercial vendors. I discovered that there are a wide variety of very well developed applications, but none of them met our requirements of portability and production. Therefore, I reluctantly began to experiment with Proc Annotate, which has basic graphic commands, allowing the user to build, piece by piece, simple diagrams and symbols in a graphical space. In addition, there are higher level graphic macros supplied both by SAS® and by SAS® developers that allow the user to construct images that are a little more complex. Descriptions of the Macros can be found at http://support.sas.com/onlinedoc/913/docMainpage.jsp under the SAS®/GRAPH reference, under ‘The Annotate Facility’, in the ‘Annotate Dictionary’, along with the basic annotate functions. While the macros described therein provide a level of functionality above that offered by the basic tools, the programmer is still responsible for almost all aspects of the graphical output.

DEVELOPMENT
All was not lost however, as the set of macros bundled with the product perform many higher level operations, such as drawing a circle or creating a pie slice.
After some experimentation, we could draw a passable speedometer face, put sectors on it, and even label the sectors. Now, how do we add two (2) arrows, one representing the state’s performance in the current year and the other representing the performance in a baseline year? And, associated with this problem, how do we translate a percentage (from 0 to 100) to the appropriate degree and position (x,y) for the end point of the speedometer needle on a 180 degree semi-circle?

Consultation with a High School geometry tutor solved the arrow location problem, but the basic tools and the SAS supplied macros to draw a needle with an arrow head were limited and aesthetically unsatisfactory for the polished product required by our client.

A very careful review of SUGI and WUSS proceedings produced the missing piece of the solution. Chung and Huang presented “%CYARROW, A NEW ANNOTATE MACRO” at SUGI 30, and the macro they developed produced an arrow that was satisfactory for our client.

PRODUCTION

Now that we had proof of concept, the production build simply involved multiple iterations and fine tuning the generated image which is characteristic of all graphic applications targeted for publishing. This is where building a batch process in SAS® really pays off.

It was difficult to define the graphic space and draw within it in such a way as to retain proportionality and meet the publishing requirements. I would like to inform the reader that it was a simple application of logic and understanding, but it was more trial and error.

Once the graphical frame was established and the graphic could be drawn with reliability, then the fine points of color combinations, labeling, and titles could be worked out in an iterative fashion.

The production was data driven from the input analysis file. I developed a series of macros that set up the output graphic file and type. We experimented with multiple output types and then settled on bitmap (.BMP) because it had the best font selection and resolution. The client had specified GIF or JPEG files, but the output from SAS®/graph in these two formats lacked the clarity needed. The .bmp files were converted in batches to GIF format using Advanced Batch Converter™. This conversion substantially reduced the size of the graphic files and allowed us to easily set the pixel width. Once the code to create a single image was stabilized, I read the analysis data base using SAS® and generated a SAS® program that iteratively called these macros to create the graphic for each state/comparison and output .bmp files with a unique name.

After several iterations and some fine tuning for specific title positions, we produced all of the speedometer graphics required. The speedometer graphics were then converted into the required web compatible format; a descriptor (alt text that describes a graphic for screen readers designed for the visually impaired) was added; and they were included in the Web application.

Following is a documented example of the code:

```sas
%let pgm=proddata;
* Output file names should be: 
* SS_Level_CompGroup_Speedo 
* where: 
* SS=STPC or "US" or "R1" to "R9" for nine regions 
* Level = 0 to 9 
* CompGroup= N for national, R for region 
* ************************************************ 
libname sas01 "C:\Documents and Settings\kenney\My Documents\My SAS Files\9.1" ;

Data speedo ;
set sas01.speedometer3 ; *data with levels ;
nspeed=Speed/100 ;
format speed 15.10 nspeed 15.12 ;
run ;
 ************************************************
* clean up values with formats 
************************************************
proc format ;
value $ STPCfmt 'RI'='Rhode Island'
```
'US'='Nation'
'R1'='Northeast'
'R2'='Midwest'
'R3'='South'
'R4'='West';

value compnfmt 3='Above Average'
2='Average'
1='Below Average';

value $ cmpgfmt 'N'='Nation'
'R'='Region';

value levlfmt 0='All Measures'
1='Preventive Measures'
2='Acute Care Measures'
3='Chronic Care Measures'
4='Home Health Care Measures'
5='Hospital Care Measures'
6='Nursing Home Care Measures'
7='Ambulatory Care Measures'
8='Diabetes - Process'
9='Diabetes - Outcome';

value relyfmt 1='Current'
2='Base';

run;

options noxwait;

%macro pickfmt(outfile=,fmttype=,outext=);

%*--------------------------------*;
%* MACRO PICKFMT                  *
%* Macro to toggle goptions       *
%* parameter values depending on  *
%* what output format we pick for *
%* the graphics data set          *
%*--------------------------------*;

%if &outext > %then %do;
  filename grafout "C:\Documents and Settings\kenney\My Documents\My SAS Files\9.1\&outfile..&outext.";
  x "erase &outfile..&outext.";
%end;
%else %do;
  filename grafout "C:\Documents and Settings\kenney\My Documents\My SAS Files\9.1\&outfile..&fmttype.";
  x "erase &outfile..&fmttype.";
%end;

goptions gsfname=grafout
gsfmode=append
device=&fmttype.;
%mend pickfmt;

**********************************************************************
* build speedos
**********************************************************************;
* application specific sizing info for SAS/GRAPH;
goptions hsize=7 vsize=3.94 hpos=30 vpos=18 ftext=swissx;
* grab the proc annotate macros;
%annomac(nomag);
* grab the Chung and Huang macro for arrows;
%include "cyArrow.sas";

%macro convrt3(rate=,colr=,xc=,yc=,linetype=);
* method to locate x,y in space based
  on percentage and given center of half circle.;
PI = constant('pi');
  cx = radius*cos(PI-&rate.*PI) + &xc;
  cy = radius*sin(PI-&rate.*PI) + &yc;
* where (&xc,&yc) is the center of the circle, and 'radius' is the radius;
* make call to arrow builder:
  %cyArrow(&xc,&yc,cx,cy,color="&colr",size=5,barbangle=50,line=&linetype.)
%mend convrt3;

* define center of semi-circle;
%let centerx=15;
%let centery=6;

* define X,Y starting positions for sector labels;
%let vwkx=%eval(&centerx.-10); * 'very' of 'very';
%let vwky=%eval(&centery.+3); * 'weak';
%let zeroy=%eval(&centery.+2);
%let zerox=%eval(&centerx-10); * 'very weak';
%let lowx=%eval(&centerx-6); * weak;
%let lowy=%eval(&centery+8);
%let midx=%eval(&centerx+2); * 'average';
%let midy=%eval(&centery+10);
%let highx=%eval(&centerx+9); * strong;
%let highy=%eval(&lowy);
%let vstrgVx=%eval(&centerx.+12); * 'very' of 'very';
%let vstrgVy=%eval(&zeroy.+1); * 'strong';
%let allx=%eval(&centerx+13); * strong;

* x position for performance label;
%let perfx=%eval(&centerx+8);

%macro makespdo(rate1=,rate2=,lvlname=,lvllen=);
  *add level name under performance label;
  %let lvllx=%eval(&centerx+&lvllen);
  * build annotate command file;
  data toano;
  %* set annotate macro values to defaults lengths and data types;
  %dclanno;
  %* except for text, which we control;
  length text $ 27;
  *create slices;
  %slice(&centerx,&centery,0,36,10,VIYG,PS,4) ; *creates a 1/10 pie in vivid yellow green;
  %slice(&centerx,&centery,36,36,10,BILG,PS,4) ; *creates a 1/10 pie in brilliant yellowgreen;
  %slice(&centerx,&centery,72,36,10,yellow,PS,4) ; *creates a 1/10 pie in yellow;
  %slice(&centerx,&centery,108,36,10,GOLD,PS,4) ; *creates a 1/10 pie in gold;
  %slice(&centerx,&centery,144,36,10,Crimson,PS,4) ; *creates a 1/10 pie in crimson;

  * write labels;
  %label(&perfx,2,'Performance Meter:',black,0,0,1.5,,1);
  %label(&lvllx,.5,"&lvlname",black,0,0,1.5,,1);
  * label sectors;
  %label( &vwkx,&vwky,'Very',black,0,0,1,,1) ; *stack 'very' over;
  %label( &zerox,&zeroy,'Weak ',black,0,0,1,,1) ; * 'Weak';
  %label( &lowx,&lowy,'Weak',black,0,0,1,,1) ;
  %label( &midx,&midy,'Average',black,0,0,1,,1);
  %label( &highx,&highy,'Strong',black,0,0,1,,1);
  %label( &allx,&zeroy,'Very Strong',black,0,0,1,,1);
  %label( &vstrgVx,&vstrgVy,'Very',black,0,0,1,,1) ; *stack 'very' over;
  %label( &allx,&zeroy,'Strong',black,0,0,1,,1) ; * 'strong';

  * creat arrows and hub;
  %frame(white, 1, 1)
  %move(&centerx,&centery) ; %*move curser to center of speedo;
  radius = 8; %*define radius;
  %move(&centerx, &centery) ;

%end makespdo;
%slice(&centerx,&centery,0,180,1,black,PS,4) ;  %* creates a 1/2 pie in black ;
style='PS';                                           %* issue a single annotate command ;
function='PIE' ;                                      %* to write a hub over the base ;
color='black' ;                                       %* of the arrows to clean up the ;
x=&centerx.;                                          %* look. ;
y=&centery.;                                          %*;
line=3 ;
angle=0;
rotate=180;
when='A';
output ; run ;
proc print data=toano ;
  title3 "level &lvlname." ;
run ;
proc ganno annotate=toano ;
run ;
%mend makespdo;

%macro aspeedo(sname=,
currentrate=,
baserate=,
ln=,
l1=)
  ;
%*------------------------------------------* ;
%* MACRO ASPEEDO                            * ;
%* Called for each rate comparison within   * ;
%* each state to produce an output file with* ;
%* speedometer image                        * ;
%* Output file has unique name.             * ;
%* Parameters:                              * ;
%*    sname=Output file name                * ;
%*    currentrate=States rate               * ;
%*    baserate=comparison rate              * ;
%*    ln=Level Name                         * ;
%*    l1=Level length for center            * ;
%*------------------------------------------* ;

%pickfmt(outfile=&sname.,fmttype=bmp) ;
%makespdo(rate1=&currentrate.,rate2=&baserate.,lvlname=&ln.,lvllen=&l1.) ;
%mend aspeedo ;

%* here is where we start using the data to drive the macro production ;
proc sort data=speedo ;
  by stpc level compgroup relyRflg ;
run ;
%macro skipspeedo(indat=) ;
%*---------------------------------------* ;
%* MACRO SKIPSPEEDO                      * ;
%*  Creates a SAS program that makes     * ;
%*  multiple calls to the speedo making   * ;
%*  macro 'aspeedo' defined above.        * ;
%*  Data driven from client file,         * ;
%*  collects 2 rates across observations * ;
%*  and spits out a call to the macro    * ;
%*---------------------------------------* ;

%* delete output file ;
%* define output file ;
data _null_ ;
set &indat. ;
length outname $20 levlnam $26 ;
retain outname c_rate b_rate ;
by stpc level compgroup relyRflg ;
outname=compress(stpc||"_"||put(level,1.)||"_"||compgroup||"_Speedo")
if relyRflg=1 then c_rate=nspeed;
else if relyRflg=2 then b_rate=nspeed;
if last.compgroup then do;
* grab name of level;
levlname=put(level,levlfmt.)
* attempt to center, font dependent;
lnamel=int(.95*{(length(trim(levlname)))/2})
* fix centering for labels that are not behaving;
if levlname='Preventive Measures' then lnamel=lnamel-1;
else if levlname='Diabetes - Outcome' then lnamel=lnamel+1;
file "C:\Documents and Settings\kenney\My Documents\My SAS Files\9.1\speedrun.sas"
put 'aspeedo(sname='outname ',currentrate=' c_rate ',baserate=' b_rate ',ln=' levlname
',ll=' lnamel ');'
outname=''
c_rate=.
b_rate=.
end;
run;

/* run output file;
%include "C:\Documents and Settings\kenney\My Documents\My SAS Files\9.1\speedrun.sas"
%mend aspeedo;
/* launch application;
%aspeedo(indat=speedo) ;

A sample of the SAS program generated by this process:
%aspeedo(sname=CA_0_N_Speedo ,currentrate=0.4210526316 ,baserate=0.51875 ,ln=All Measures
,ll=5 )
%aspeedo(sname=CA_0_R_Speedo ,currentrate=0.399122807 ,baserate=0.46875 ,ln= All Measures
,ll=5 )

A sample of the input data:

<table>
<thead>
<tr>
<th>Obs</th>
<th>Line</th>
<th>Angle</th>
<th>Rotate</th>
<th>Size</th>
<th>Radius</th>
<th>PI</th>
<th>CX</th>
<th>CY</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
<td>36</td>
<td>10.0</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>Performance Meter:</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>36</td>
<td>36</td>
<td>10.0</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>72</td>
<td>36</td>
<td>10.0</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>108</td>
<td>36</td>
<td>10.0</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>144</td>
<td>36</td>
<td>10.0</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>1.5</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td></td>
</tr>
</tbody>
</table>

CONCLUSION
If you have a very well defined graphical application that must be duplicated with minor changes a large number of times, a satisfactory custom production system can be built using BASE SAS®, SAS® Macros, and the Annotate procedure in SAS®.
REFERENCES

ACKNOWLEDGMENTS
A team lead by Dwight McNeill, AHRQ (Dwight.McNeill@ahrq.hhs.gov) including Rosanna Coffey, Thompson Medstat (Rosanna.Coffey@thomson.com), Craig Hunter, Thompson Medstat (Craig.Hunter@thomson.com), Marguerite Barrett, M.L. Barrett, Inc. (barrettml@earthlink.net), and Josh Cassell, Geometry tutor (joshcassell04@hotmail.com).

RECOMMENDED READING
http://support.sas.com/events/sasglobalforum/previous/online.html - SUGI proceedings.

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

Author Name       Timothy A. Kenney, MA
Company            Kenney IS Consulting, Inc.
Address            109 N Montgomery St., Suite D
City state ZIP     Ojai, CA       93023
Work Phone:        (805) 646 9982
Email:             tkenney@west.net

SAS® and all other SAS® Institute Inc. product or service names are registered trademarks or trademarks of SAS® Institute Inc. in the USA and other countries. © indicates USA registration.

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