Enhancing SAS/GRAPH® Output with the Annotate Facility

Steven First
Systems Seminar Consultants

Abstract
The SAS/GRAPH Annotate facility allows the user to enhance graphics created with SAS/GRAPH software. This paper presents the basic concepts of the Annotate facility and introduces the audience to text placement, symbol placement, line drawing, polygon construction, map labeling, custom legends, and line types. Also covered are SAS Institute provided annotation macros and advanced functions to allow complex labeling and freehand graphics.

Introduction
When SAS/GRAPH Version 5 software was released, the new Annotate facility provided a comprehensive solution to some long requested missing capabilities with the product. Annotation is any text, symbol, line, or polygon that is overlaid on an existing graphic image. There are at least three or four different levels of annotation required.

First a simple method was needed to place annotation anywhere on a graphic image. An example of this type of annotation might be to place a company logo in the same place on all graphs regardless of the titles and footnotes a user chooses.

A second need was to place annotation in the plot "window". This would allow the user to code as many titles and footnotes as needed again, but have the annotation appear below the titles and above the footnotes.

A third need was to somehow tie the annotation to the data axes. An example of this is to place text that "floats" above the bars of a chart. This is the most complex type of annotation because the data values might change from run to run and the annotation must move with the graph.

Finally it would be useful to be able to generate annotation from another source. An example of this might be to read a text file, and generate text charts using SAS/GRAPH fonts and colors.

The Annotate facility does all of the above and much more. It uses a rather batch oriented approach and does not offer features like interactive cursor draw at this time.

Ways to modify SAS/GRAPH output
Without annotation we can enhance graphic output by:

- Changing PROC options or additional statements
- Changing the data
- Using titles, footnotes, notes
- PROC GREPLAY can place multiple pictures on a page

The Annotate facility allows the user to:
- Place text anywhere on the screen
- Draw lines or other figures
- Define polygons to fill and color
- Create custom "free hand" graphics
- Place text or images on maps

The annotate commands are stored in a SAS dataset using special variables. This dataset can be specified on the ANNOTATE= parameter for PROCs GANNO, GCHART, GCONTOUR, G3D, GMAP, GPLLOT, and GSLIDE. When an annotate dataset is specified the PROC will draw the graphic as usual, and then overlay the annotation on the graph. PROC GANNO produces no graphics by itself and is intended to produce only annotation.

The following program will create a simple annotate dataset and then draw the image shown in figure 1. Note that the variables must be named exactly as shown. Many other variables can be added, but if not present a default value will be used.

```
DATA ANNOTATE;
  INPUT FUNCTION $ X Y SIZE TEXT $20.;
  DATALINES;
  MOVE 20 10
  DRAW 20 30 5
  DRAW 80 30 5
  DRAW 80 10 5
  DRAW 20 10 5
  LABEL 50 30 $• A simple box
; PROC GANNO ANNOTATE=ANNOTATE; RUN
```

Figure 1 A simple box
Figures 2 through 6 show various capabilities of the annotate facility.

Figure 2 Various polygons

Figure 3 Map annotation

Figure 4 Map annotation

Regional Sales

<table>
<thead>
<tr>
<th>Region</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>North</td>
<td>$244,678</td>
</tr>
<tr>
<td>South</td>
<td>$532,123</td>
</tr>
<tr>
<td>East</td>
<td>$564,343</td>
</tr>
<tr>
<td>West</td>
<td>$167,343</td>
</tr>
<tr>
<td>Alaska</td>
<td>$233,434</td>
</tr>
<tr>
<td>Hawaii</td>
<td>$171,232</td>
</tr>
</tbody>
</table>

Figure 5 Text applications

Figure 6 Freehand graphics

Annotation sequence

The following steps are required to annotate a graph:

1. Identify the components of the image.
2. Establish a frame or reference.
3. Digitize components, functions, attributes.
4. Store the above as variables in a SAS dataset.
5. Run a SAS/GRAPH PROC with ANNOTATE option.

All components images must be defined on a cartesian coordinate system as shown in figure 7.
Annotation variables

Special Annotate variables can be grouped into five categories:

1. To establish a frame of reference.
2. To set a function or identify a component.
3. To position an image.
4. Enhance the image display.
5. Position text.

Each of the above categories will control a different part of annotation. The dataset can contain more than the special variables in which case they will be ignored. Some examples and values of some of the variables are:

<table>
<thead>
<tr>
<th>Category</th>
<th>Variable</th>
<th>Type</th>
<th>Length</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XSYS</td>
<td>char</td>
<td>1</td>
<td>'4'</td>
</tr>
<tr>
<td></td>
<td>YSYS</td>
<td>char</td>
<td>1</td>
<td>'4'</td>
</tr>
<tr>
<td>2</td>
<td>FUNCTION</td>
<td>char</td>
<td>1</td>
<td>'LABEL'</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>num</td>
<td>8</td>
<td>XLEFT</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>num</td>
<td>8</td>
<td>YLEFT</td>
</tr>
<tr>
<td>4</td>
<td>SIZE</td>
<td>num</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>STYLE</td>
<td>char</td>
<td>6</td>
<td>'NONE'</td>
</tr>
<tr>
<td></td>
<td>COLOR</td>
<td>char</td>
<td>6</td>
<td>1st in device list</td>
</tr>
<tr>
<td></td>
<td>LINE</td>
<td>num</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>POSITION</td>
<td>char</td>
<td>1</td>
<td>'5'</td>
</tr>
<tr>
<td></td>
<td>TEXT</td>
<td>char</td>
<td>200</td>
<td>blank</td>
</tr>
</tbody>
</table>

It is critical that the variables are created with the correct attributes. It is a very good idea to use PROC CONTENTS and PROC PRINT to display the annotate dataset at least while debugging because the annotate facility is very fussy about variable attributes and there is no point trying to annotate graphics if the underlying dataset is incorrect.

The program below uses default annotation to display the text shown in figure 8.

DATA DEFAULTS;
  INPUT X Y TEXT 9 50.;
DATA LABELS;
  50 30 Label is the default function
  50 20 with text centered about X and Y

Figure 8 A default annotation

Frames of reference

The Annotate facility can address three areas of the screen: the entire screen, the window, or the same data coordinate system that the PROC uses. All of the frames can be referenced by percentage or by screen cells, and can be referred to in a relative or absolute manner. Figure 9 shows the three frames of reference.

Figure 9 Frames of reference

The desired frames of reference are chosen by setting variables called XSYS and YSYS as shown in the table below. Note that the two variables don't need to use the same frame of reference.

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>UNIT</th>
<th>AREA</th>
<th>VALUE of XSYS, YSYS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSOLUTE</td>
<td>DATA X</td>
<td>0 to 100 % of AXIS area</td>
<td>X1</td>
</tr>
<tr>
<td></td>
<td>DATA</td>
<td>AXIS MIN to AXIS MAX</td>
<td>X2</td>
</tr>
<tr>
<td></td>
<td>SCREEN</td>
<td>0 to 100 % of SCREEN</td>
<td>X3</td>
</tr>
<tr>
<td></td>
<td>SCREENCELL</td>
<td>0 to width of SCREEN</td>
<td>X4</td>
</tr>
<tr>
<td></td>
<td>WINDOW</td>
<td>0 to 100 % of WINDOW</td>
<td>X5</td>
</tr>
</tbody>
</table>

206
Using '5' as a value in XSYS and YSYS tells the annotate facility to use a percentage of the window area and allows annotation between titles and footnotes. The following program created figure 10.

```
DATA WINDOW;
  INPUT $XSYS $ YSYS $ $ 50;
FARTAILINES;
  $ $ $ This line is directly under titles
  $ $ $ this is the center of the window
  $ $ $ this is about just above footnotes
PROC SIDEBAR ANNOTATE-WINDOW;
TITL $ CHED FORMATS 'Annotation In';
FOOTNOTE $ CHED $ FORMATS X2 = 'The window';RUN;
```

Figure 10  Percentage of windows

A value of '3' in XSYS and YSYS tells the Annotate facility to use a percentage of the screen frame of reference. This allows the user to refer to the same data area that the procedure will be using. The example shown in figure 12 is a PROC GPLOT which has Y axis values (RATE) between 0 and 20, and X axis values (YEAR). The program below moves those same values to X and Y in the annotate dataset which allows the user to position text directly over the point on the plot. This particular example uses one dataset for the plotting and another for the annotation, but it could be easily modified to generate the annotation from the plot data using SET instead. Note also that PROC GPLOT and PROC GCHART can also plot character variables. If this were the case, the Annotate facility requires that the character variables XC and YC be used instead of X and Y.

```
DATA MORTGAGE;
  INPUT $XSYS $ YSYS $ $ 50;
FARTAILINES;
  3 3 50 16 Annotated mortgage plot
  3 3 25 05 This example uses two frames of reference
  2 2 1978 12 (1978,12)
  2 2 1980 14.1 (-980,14.1)
  2 2 1982 15 (1982,15)
  2 2 1984 13.6 (1984,13.6)
  2 2 1986 9.7 (1986,9.7)
PROC GPLOT DATA=MORTGAGE;
  PLOT RATE * YEAR;
  MAXAXES=978 TO 1986 BY 2
  VAXIS=0 TO 20 BY 2
  ANNOTATE=MORTG;
  SYMBOL 1 =JOHN Y=NONE;
  TITLE * *;
  FOOTNOTE * *;
RUN;
```

Figure 11  Percentage of screen

A value of '2' in XSYS and YSYS selects the data frame of reference. This allows the user to refer to the same data area that the procedure will be using. The example shown in figure 12 is a PROC GPLOT which has Y axis values (RATE) between 0 and 20, and X axis values (YEAR). The program below moves those same values to X and Y in the annotate dataset which allows the user to position text directly over the point on the plot. This particular example uses one dataset for the plotting and another for the annotation, but it could be easily modified to generate the annotation from the plot data using SET instead. Note also that PROC GPLOT and PROC GCHART can also plot character variables. If this were the case, the Annotate facility requires that the character variables XC and YC be used instead of X and Y.
Additional annotation variables

The plot above centers the text directly over the plot points and since the plot line and axis also are drawn at those points, the annotation is cluttered. We need a method to move the annotation slightly away from the lines and the axis. In addition we may need to place other images such as lines, bars etc on the graph. The Annotate facility allows several other values of the FUNCTION variable to do the placing as shown in the table below. There are more advanced functions as well that are not listed here.

Values for FUNCTION:

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAB</td>
<td>Placement text at X,Y coordinates</td>
</tr>
<tr>
<td>MOVE</td>
<td>Move with the pen up</td>
</tr>
<tr>
<td>BAR</td>
<td>Construct and fill a rectangle</td>
</tr>
<tr>
<td>DASH</td>
<td>Dash pen to close polygon</td>
</tr>
<tr>
<td>POINT</td>
<td>Place a simple point at X,Y</td>
</tr>
<tr>
<td>POLY</td>
<td>The beginning of a polygon to fill</td>
</tr>
<tr>
<td>SYMBOL</td>
<td>Place a symbol on graph</td>
</tr>
</tbody>
</table>

The LABEL FUNCTION depends on several other variables to help place the text and also specify other attributes such as font, color, angle, etc. Below is listed some of the associated variables and their usage.

POSITION Value 0 to 11 (default = 5)
SIZE Any positive number (default = 1)
ANGLE -90 to 90 (default = 0)
STYLE Any valid font (default = "NONE")
COLOR Any valid color (default = 1st color list)
TEXT Any text string (default = "")

The POSITION variable is especially important to align a block of text many different ways around a point. The following list shows the different options for text placement and should help us solve the placement problem in the last graph.

1. Centered, left aligned
2. One cell below, right aligned
3. One cell below, centered
4. One cell below, left aligned
5. Half cell above, right aligned
6. Half cell above, centered
7. Half cell above, left aligned
8. Half cell below, right aligned
9. Half cell below, centered
10. Half cell below, left aligned

By using the POSITION, SIZE, and STYLE variables we can enhance the GPLOT job from earlier. The label marked (1978,12) uses a POSITION value of 9 which aligns the left edge of the text one cell below the X and Y value. Now the label is positioned in available space instead of overlaid on the axis as before. Using a $ character from the GREEK font also plots the $ character over the points on the line. The program below produces figure 13.

```
DATA MORTG; 
  INPUT XS (S 2.2) YS (S 2.2) POSIT1ON S.16 STYLE S SIZE T TEXT $;
  DATALINES;
  3 3 50 96 5 "SIMPLEX 2 Annotated mortgage plot"
  3 3 5 5 "SIMPLEX 1 Left alignment is ok"
  2 2 1978 12.0 9 "GREEK 1"
  2 2 1982 14.1 7 "SIMPLEX 1 (1982,14.1)"
  2 2 1982 14.1 8 "GREEK 1"
  2 2 1982 15.2 3 "SIMPLEX 1 (1982,15.2)"
  2 2 1986 15.5 5 "GREEK 1"
  2 2 1986 13.8 3 "SIMPLEX 1 (1986,13.8)"
  2 2 1986 13.8 5 "GREEK 1"
  2 2 1986 9.7 7 "SIMPLEX 1 (1986,9.7)"
  2 2 1986 9.7 5 "GREEK 1"
...
```

Figure 13 Using POSITION, SIZE, STYLE

Line drawing

MOVE and DRAW as FUNCTION values along with the COLOR, SIZE, and LINE variables can be used to draw lines and in effect create custom interpolation types. In the below program part of the lines are drawn in a dashed pattern and the other parts solidly producing figure 14.
Another variation is to let GPLOT do the line drawing, but to use annotation to place the text "MORTGAGE" and "PRIME" near the lines to act as a legend as in figure 15.

DATA MORTGAGE;
  INPUT RATE YEAR $2.3 FUNCTION $ Y X Y;
  POSITION $ 2 STYLE $ SIZE LINE TEXT $;
  DATALINES;
  12.0 1978 MORTGAGE
  9.1 1978 PRIME
  16.1 1980 MORTGAGE
  13 1980 PRIME
  1982 MORTGAGE
  16 1982 PRIME
  13.8 1984 MORTGAGE
  13 1984 PRIME
  9.7 1985 MORTGAGE
  8.7 1985 PRIME

DATA LABELS;
  X$="1978";Y$="1978";POSITIO$="11";
  STYLE="SIMP\[E\]";TEXT="MORTG\[A\]GE";COLOR="BLUE";
  OUTPUT;
  Y%'I';TEXT="PRIME";COLOR="RED";
  OUTPUT;

PROC GPLOT DATA=MORTGAGE;
  PLOT RATE * YEAR=TYPE/OVERLAY NOLEGEND
  NOLEGEND MAX=1978 TO 1986 BY 2
  VAXIS=O TO 20 BY 2 NOANNOTATE;
  SYMBOL l=JOIN COLOR=BLUE;
  PATTERN1 V=SLANT C=RED
  PATTERN2 V=SLANT C=BLUE;
  TITLE1 FONT=USIS C=BLUE "Mortgage"
  FONT=USIS C=RED "Prime rate";RUN;

Figure 16  A custom legend

CUSTOM LEGENDS:

Another use of the Annotate facility is to build custom legends in available space. The BAR or PIE value for FUNCTION can draw rectangles or circles, and the LABEL FUNCTION can add the needed text as shown in figure 16.

DATA LEGENDS;
  LENGTH TEXT $16 FUNCTION STYLE $; WHEN='A';
  INPUT X$ Y$ Z $ FUNCTION $ X Y;
  VAXIS=O TO 20 BY 2
  NOANNOTATE;
  SYMBOl$; JOIN V=NONE C=RED;
  PATTERN1 V=SLANT C=BLUE;
  PATTERN2 V=SLANT C=RED;
  TITLE1 FONT=USIS C=BLUE "Mortgage"
  FONT=USIS C=RED "Prime rate";RUN;
Bar chart annotation

Another common application is to place annotation at the end of a bar chart. This is easy to do by again using the data frame of reference. The position variable can place the text above the bar as shown, or if desired position could be used to place text within the bar. Again the data frame of reference allows the annotation to "float" on the end of the bars, and the program could be easily written such that the annotation will automatically move if different data is input. The Annotate facility doesn't support any "clipping" so unless the text has a contrasting color, it may not show very well and depending on the device it may smear if drawn before the bar is filled. The WHEN variable can be set to 'A' if the annotation should take place after the chart is draw, and a value of 'B' will draw annotation before the chart drawing. The following program produces the output shown in figure 17.

```
DATA MORTBAR:
  INPUT XSYS 1;
  2 2 1978 12.0 2 SIMPLEX 1 12.0
  2 2 1980 14.1 2 SIMPLEX 1 14.1
  2 2 1982 15.0 2 SIMPLEX 1 15.0
  2 2 1984 13.8 2 SIMPLEX 1 13.8
  2 2 1986 9.7 2 SIMPLEX 1 9.7;
PROC GCHART DATA=MORTBAR;
  SERIES=REGION
  MDSM=10 15 BY 2
  ANNOTATE=MORTBAR.
  PATTERN1 CHED=WHITE;
  TITLE1 CHED=WHITE 'Home mortgage rates';
  TITLE2 CHED=WHITE 'Annual percentage rates';
RUN;
```

```
DATA SALES:
  INPUT REGION 8:6;SALES;
  LENGTH TEXT 50;
  RETURN TEXT TEST XSYS \S 1 0 0;
  VDEC=70/6;
  SIZE=VDEC*0.5;
  TITLE=\SISU ';
  COLOR=BLACK ';
  GROUP=';
  X=15;Y=YDEC;
  TEXT=REGION;OUTPUT;
  PATERN1 CHED=WHITE;
  COLOR=\SISU ';
  POSITION=';
  X=85;
  TEXT=SALES, DOLLAR12,;);
OUTPUT;
DATA ATLAS:
  North 264476
  South 532123
  East 564313
  West 167313
  Alaska 233413
  Hawaii 171232
PROC GCHIDE ANNOTATE=SALES;
  TITLE='Regional Sales';
RUN;
```

```
Figure 17  Annotated bar chart

Text applications

A good application of the Annotate facility is to build text charts. The user has all the text features available and is able to build attractive word charts easily. This program uses a frame of reference that specifies '3' or percentage of screen. Note that the program also has a variable called HSYS which is also set to '3'. This variable is used to set the size of the text string as a percentage of screen. The program uses a TITLE statement, but then assumes 70% of the screen is left for the remaining text lines. The Y variable starts at 90 and X starts at 15. After a text line is output, a value called YDEC is used to move down the graph. YDEC is defined as 70 percent of the screen divided by the number of lines to display (6). Each line has a size of half the YDEC value and finally the dollar value is placed at horizontal position 85. Even though this application uses hard coded positions, systems can be easily created to generate word charts from input text files and use all the SAS/GRAPH text features. The program below generated figure 18.

```
DATA ATLAS:
  INPUT REGION 8:6;SALES;
  LENGTH TEXT 50;
  RETURN TEXT TEST XSYS \S 1 0 0;
  VDEC=70/6;
  SIZE=VDEC*0.5;
  TITLE=\SISU ';
  COLOR=BLACK ';
  GROUP=';
  X=15;Y=YDEC;
  TEXT=REGION;OUTPUT;
  PATERN1 CHED=WHITE;
  COLOR=\SISU ';
  POSITION=';
  X=85;
  TEXT=SALES, DOLLAR12,;);
OUTPUT;
DATA ATLAS:
  North 264476
  South 532123
  East 564313
  West 167313
  Alaska 233413
  Hawaii 171232
PROC GCHIDE ANNOTATE=SALES;
  TITLE='Regional Sales';
RUN;
```

```
Figure 18  Text applications

Map annotation

Another good annotation application is to place text or other annotation on maps. SAS/GRAPH provides several maps, and also a dataset called USCITY containing the names of major cities in the US. This dataset is already projected into X and Y coordinates
and can be easily used to annotate any US map if it is projected. The following example creates an annotate dataset containing the major cities in Wisconsin. It then creates a small map of Wisconsin from the SASMAPS.US dataset which includes the entire country. A simple PROC GMAP with the ANNOTATE option is all that is needed to draw the map shown in figure 19.

```sas
DATA WfCJTIU,;
SEX TEXT40;
LENGTH COLN 7IU 5.; SHU S. 8;
SASMAPS.US;
IF STAt:E='STFS:':TSlS..Z;
TEXT= "LABEL";
POSITION= 'A';
COLOR='RED';
STYLE='S':FLFLEX';
OUTPUT;
....
DATA WfUNPROJ;
SEX SASUS-
STATE $;
STATE= 55;
FR1:fRl.F='US.';
....
DATA ALL;
壹 DATA WfUNPROJ;
STATE STAt:E/PfOJECT=
MAP='WfCJTIU';
TITLE MED 'F-XSUl$S 'uisc_;n Cit;fls t ;
NOTE 'Source: Projected States dataset';
PATTERN COLOR=BLACK V=EMPTY;
RUN;
```

**Figure 19** Projected annotation

Annotating unprojected maps is a much more difficult task. The SAS/GRAPH unprojected maps provide much more detail and generally look better. The key to the following program running successfully is that the annotation and the map must be projected at the same time. This means that the two datasets must be combined, projected, split apart, and then PROC GMAP can draw the final map. The coordinates themselves need to be transformed before projection using a rather obscure formula dug out of the SAS/GRAPH User's Guide. The following program produced figure 20.

```sas
DATA WfCJTIU,;
SEX TEXT40;
LENGTH COLN 7IU 5.; SHU S. 8;
SASMAPS.US;
IF STAt:E='STFS:':TSlS..Z;
TEXT= "LABEL";
POSITION= 'A';
COLOR='RED';
STYLE='S':FLFLEX';
OUTPUT;
....
```

**Figure 20** Unprojected annotation

SAS/GRAPH comes with an extensive library of annotate macros that can be used to simplify setting up annotation datasets. They are normally stored in the SAS sample source library in a member called ANNOMAC. If they are not in a library that is included in the SAS autocall library list, the ANNOMAC member must be %INCLUDEd before they will be available for use. They are rather lengthy so you may want to turn off SOURCE2 before including them. It should be noted that these macros should be used only in a DATA step. A list of the macros and their usage is described below.

- **XCLNGTH** Set length and data type of variables
- **XCOMMENT** Insert comments
- **XSEQUENCE** Draw before or after other graphics proc
- **XSYSTEM** Define frame of reference
- **XMOVE** Move with pen up
- **XDRAW** Move with pen down
- **XLINE** Draw a line between two points
- **XLABEL** Add text
- **XRECT** Draw an empty rectangle
- **XBAR** Draw a filled rectangle
- **XCIRCLE** Draw a circle
- **XSLICE** Draw a slice
- **XENTRY** Locate a point on previous slice
- **XPOLYGON** Start a polygon
- **XPOLYCONT** Continue a polygon
- **XSCALE** Scale coordinates
- **XPU** Push text and control onto LIFO stack
- **XPO** Pop text and control onto LIFO stack
- **XMAP** Exchange text and control coordinates
- **XTEXTCONT** Assign text coordinate to control coordinate

211
Miscellaneous applications

An application of the annotate macros might be to generate a grid of the cells for a graphics device. The program below created figure 21 for a device that has 40 rows (VPOS) and 100 columns (HPOS). It could be easily modified to any HPOS or VPOS, or it might use another frame of reference such as percentage of screen and draw lines every 10% or so. If the grid is drawn on transparency material it can be overlaid over any graph and can be used to help place annotation.

DATA GRID;
DECLARE :
DO Y=0 TO 40;
  //LINE(Y,100,Y,BLACK,1,1);
END;
DO X=0 TO 100;
  //LINE(X,0,X,40,BLACK,1,1);
END;
RUN;
PROC GLIDE ANNOTATE=GRID;
TITLE;
RUN;

Figure 21 A generated grid

Another option may be to use a similar grid and have the annotate facility overlay it on a chart. By drawing a grid with lines every 10% and overlaying it on the chart, the user can note the locations of needed annotation and enter it in an annotate dataset which can be used to produce figure 22.

DATA ANNOTES;
DECLARE;
XSYS=5';TEXTS='5';HYSYS='4'; * USES SCREEN PERCENTAGE ;
WHERE='A';
IF _X=1 THEN 
DO Y=0 TO 100 BY 10;
  //LINE(Y,100,X,BLACK,1,1);
END;
DO X=0 TO 100 BY 10;
  //LINE(X,0,X,40,BLACK,1,1);
END;
RUN;
PROC PRINT DATA=ANNOTES;
TITLE 'STANDARD GRID DATASET';
RUN;

Figure 22 Bar chart with grid

After looking at the previous graph an annotate dataset can be built and the final graph drawn as in figure 23. This technique is quick if only one graph needs to be annotated, but if the data vary from run to run, it may be better to use the data frame of reference as described earlier.

DATA ANNOTES;
DECLARE;
XSYS=5';TEXTS='5';HYSYS='4'; * USES SCREEN PERCENTAGE ;
WHERE='A';
IF _X=1 THEN 
DO Y=0 TO 100 BY 10;
  //LINE(Y,100,X,BLACK,1,1);
END;
DO X=0 TO 100 BY 10;
  //LINE(X,0,X,40,BLACK,1,1);
END;
RUN;
PROC PRINT DATA=ANNOTES;
TITLE 'STANDARD GRID DATASET';
RUN;

Figure 23 Bar chart with grid
Freehand graphics

Freehand graphics can be drawn by following the following steps:

1. Include all points in all polygons.
2. Use MOVE, DRAW to draw lines.
3. Use LABEL for drawing text.

This can be a very time consuming task if done manually, but if the polygons and text are already in a machine readable form it can be automated. The dataset below defines four different polygons and the text on the graph. It actually consists of several hundred points, but all were not shown because of space. The final graph is shown in figure 24.

DATA CYCLONE;
    DECLARE;
    PROC PRINT, CONTENTS;
    TITLE "DATA CYCLONE";
    TITLE "INPUT X 1-5 Y 7-11 FUNCTION 6:1-0 TITLE 82-29";
    COLOR 931-30 637 TEXT 53-56 POSITION 55 LINE 57;
    DATALINES;
    100 100 POLY KSAS BLACK RIGHT CLOUD 18.66 91.41 POLY MCKSAS BLACK;
    79.65 98.15 POLY MCKSAS BLACK;
    71.41 100 POLY KSAS BLACK CYCLONE 79.65 98.15 POLY MCKSAS BLACK CYCLONE;
    100 100 POLY KSAS BLACK UP LEFT CLOUD 18.66 98.44 POLY MCKSAS BLACK;
    32.93 67.74 MOVE;
    57.06 67.74 DRAW;
    57.55 67.74 LABEL SIMPLEX BLACK (1985) 005;
    23.7 77.65 LABEL SIMPLEX RED TEXAS 026;
    31.41 40.61 LABEL SIMPLEX RED MEXICO 026;
    39.41 40.11 MOVE;
    69.72 55.16 LABEL SIMPLEX RED (1985) 026;
    proc g3d;
    annotate=cyclone;
    title=TITLE "TORNADO WATCH";
    footnote=RUN;

In summary I hope that this gives some insight to the uses and coding of the annotate facility. I have found it to be comprehensive though not always as easy to use as I would like. I have found that PROC PRINT and PROC CONTENTS of all annotate datasets helps debugging considerably. It would be a great advantage to have some cursor draw and interactive annotation capabilities in addition to the annotate facility, and hopefully SAS Institute will implement more enhancements in the future.

The author will be glad to answer questions and accept suggestions at the following address:

Steven First
Systems Seminar Consultants
6014 Gateway Green
Madison, WI 53716
(608) 222-7081

SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc.