THE SAS/GRAPH® ANNOTATE FACILITY
Anthony L. Friebel, SAS Institute Inc.

ABSTRACT
Since its introduction in 1980, users have sought ways to create specialized graphic output with the SAS/GRAPH product. Some have been more successful than others. While creative manipulation of data can often produce the desired picture, a major criticism of the SAS/GRAPH product has been its inflexibility in dealing with descriptive annotation. This paper will explore new options for customizing standard SAS/GRAPH output, as well as new SAS/GRAPH applications.

INTRODUCTION
The ANNOTATE facility permits you to customize SAS/GRAPH procedure output from variables and values contained in non-graphic SAS® data sets. The facility is capable of defining polygons, controlling pen movement, placing text on the page, and so on. It adds to existing procedures the capability of producing custom graphic output from a DATA step.

This paper is intended only as an overview of the facility, some of its capabilities and its limitations. Complete documentation is contained in Technical Report P-12B; Changes and Enhancements in the Base SAS and SAS/GRAPH Products under VMS, Version 4.

THE PROBLEM
Until now, placement of additional graphics on a given plot, whether simple lines, special symbols or a more complex application, was tedious at best. Clever DATA steps were created, or many attempts made until the proper '.M=' and '.O=' combinations were attained. Although capable of supplying a wide range of customization, one change in the data could necessitate several more attempts.

If you were clever you might achieve the desired result. Variables could be specially formatted, or text moved to obtain a specific look. But, only the simplest modifications could be made in this manner. The more complex the application, the less likely creative formatting would accomplish the effect.

The ANNOTATE facility
The ANNOTATE facility differs from these methods in that it can be made 'automatic'. With the ANNOTATE facility, additional graphics may be placed on plots in relation to actual data values, screen cell coordinates or any mix of the two. Additional special systems permit absolute or relative coordinates and percentage of area specification.

ANNOTATE, in a broad sense, may be considered a controlled access path to the graphics subroutine library. It permits you to specify commands to the graphics procedure for interpretation, execution, and eventual display on the generated plot.

The commands are contained in a SAS data set. Special variables in the data set pass information to ANNOTATE for interpretation and execution. A generalized flow of control could be visualized as below:

You may annotate your graphs by requesting the desired commands (via designated variables) in a SAS data set, that is then specified as the ANNOTATE= option to the graphics procedure.

* e.g.
PROC GMAP
DATA = sample.data
MAP = sample.map
ANNOTATE = company.logo;
ID id;
CHORO id / discrete nolegend
ANNOTATE = map.detail;
RUN;

The ANNOTATE= option is available in both the PROC and the statement lines. The data set specified in the PROC statement will be applied globally to all interior statements in the procedure. This is useful in applications where a company logo is to be placed on each plot generated. The statement specification is active only for the statement which invoked the ANNOTATE= option, and its output is placed on the graph in addition to any ANNOTATE output generated from the PROC statement data set.

Each ANNOTATE data set in use is processed twice in generation of its output—once, before any graphic output is produced by the procedure, and again after all output has been generated. This division of processing permits...
you to place annotations on the graph in relation to the order of SAS graphics generation. If labelling PROC GGRAPH area fills, for example, you will want to place the text 'AFTER' the area is drawn on the graph. Otherwise, the text will be covered up when the area is filled in.

APPLICATIONS

The special applications needs of SAS/GRAPH users may now be addressed through a common interface method within the SAS environment. The immediate uses of the ANNOTATE facility center around accurate data relative positioning of text labels, particularly in mapping applications. ANNOTATE uses may include:

- Placement of city names on maps
- Drawing of lines between points
- Labelling termination of curves
- Highlighting data minimum, maximum, or inflection points
- Special presentation graphics

The facility is designed to be flexible enough that an entire graphics procedure may be generated from a DATA step, yet sufficiently simple to profit non-programmers.

SUPPORTED FUNCTIONS

The ANNOTATE facility carries out its instructions through a user-specified series of commands and parameters in a SAS data set. Commands are the observations in the data set, parameters are the particular variables in the observation. The parameters (variables) have special reserved names which are indicative of their use.

A simplified description of an ANNOTATE observation is:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>The X or horizontal coordinate</td>
</tr>
<tr>
<td>Y</td>
<td>The Y or vertical coordinate</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>Particular feature you want to accomplish</td>
</tr>
<tr>
<td>other variables</td>
<td>Additional information</td>
</tr>
</tbody>
</table>

X/Y Coordinates

X and Y specify the coordinates on the graph to which the FUNCTION will be applied. They may be considered in any of a number of available reference systems. These coordinate systems translate/transform user-specified values to the appropriate screen locations.

Panel two (2) illustrates the reference systems currently available in the ANNOTATE facility. For our discussion here, let's assume that percentage ranges are 0% (minimum) to 100% (maximum), and non-percentage ranges are displayed data minimums and maximums. Odd numbered systems will be considered 'value systems' or 'non-percentage systems'.

System numbers one (1) and two (2) are the data systems. The shading in panel two indicates the plot area referenced by these systems. Note that the 'data area' is enclosed by the axes lines. For PROC GSLIDE, which does not have distinct data areas, the data system is the same as the window system described below.

System numbers three (3) and four (4) are the screen systems. The shading in panel two indicates the plot area referenced by these systems. These systems most closely resemble the .M= and .D= values currently used by the GSLIDE procedure and TITLE, FOOTNOTE, and NOTE statements.

System numbers five (5) and six (6) are the window systems. The shading in panel two indicates the plot area referenced by these systems. These systems are defined as the previously described screen systems, less the area required by TITLE and FOOTNOTE statements. If no TITLE or FOOTNOTE statements occur in a given procedure, then the window systems are identical to the screen systems.

All of the systems numbered one (1) through six (6) are absolute, that is, a given value is measured from the minimum value. Separate systems exist in which a value is measured from the last point referenced. To obtain the values for these relative systems, we add six (6) to the absolute system values, using hexadecimal notation ( 10='A', 11='B', 12='C' ) when the value exceeds nine (9). Using this numbering, the relative screen value system is 'A' (4 + 6 = 10 = 'A').

These reference systems may be used in any combination to specify screen locations in the ANNOTATE data set. The (X,Y) coordinate pair is internally transformed by the ANNOTATE facility to make the requested location to its formal screen designation. The X and Y variables need not be referenced with the same system value. X can be referenced as 'DATA VALUE' and Y referenced as 'window percentage' in one observation, and both coordinates as 'screen value' in the next.

The ANNOTATE facility internally tracks two coordinate pairs through processing of observations to assist in this transformation process. These values are obtained from specified (X,Y) pairs and the SYSTEM specifications that accompany them in each ANNOTATE observation. These internal variables are used in supplying default values for missing values of X or Y in subsequent observations or as additional information specifiers. Some commands use these internal coordinates as start points.
These internal pairs are designated (XLAST, YLAST) and (Xlstt, Ylstt). The pair (XLAST, YLAST) refers to the last position of the "pen" at the completion of a non-text related function (like MOVE or DRAW), while (Xlstt, Ylstt) tracks the last position of the "pen" during the LABEL function. The coordinate pairs are updated automatically and may be manipulated by the utility type commands (like PUSH or POP) available in the ANNOTATE facility, but are not directly user-specified.

Below is a list of some of the available ANNOTATE commands and a short description of each.

**COMMENT**
Performs no actual function, but is provided as a documentation aid. Text of the comment is carried in the TEXT variable.

**MOVE**
Permits movement to a specific point on the graphics page without the drawing of a line. MOVE is most often used to prepare for a DRAW command, and updates the (XLAST, YLAST) coordinates.

**POINT**
Special MOVE/DRAW to a specific point on the graphics page. On a CRT device, POINT illuminates a single screen pixel. POINT updates the (XLAST, YLAST) coordinates.

**DRAW**
Used to draw a line on the graphics page. Segment drawn is defined as from (XLAST, YLAST) to the (X, Y) specified in the DRAW command. You have control over color, style and thickness of the drawn line. DRAW updates the (XLAST, YLAST) coordinates.

**LABEL**
Places text on the graphics page. Users may specify the color, size, font, base angle and rotation of the characters displayed. The LABEL function updates the (Xlstt, Ylstt) coordinates.

**POLY**
The POLY function designates the beginning of a POLYGON DEFINITION in the ANNOTATE observation stream. Users may define the fill pattern and color, as well as the perimeter line type in this observation. The POLY function is used in conjunction with the POLYCONT function to define and fill areas on the graphics page. POLY and POLYCONT do not update the (XLAST, YLAST) coordinates.

**POLYCONT**
Successive points in the POLYGON DEFINITION are specified in separate continuation observations. The perimeter line color is specified in the first POLYCONT observation. Polygon definitions may not be interrupted by other functions. A definition is terminated by a new POLY command or by any other non-POLY function. Appropriate messages are issued by the ANNOTATE facility in the event of interruption.

**PIE**
Facilitates the drawing of pie slices on the graphics page. You may specify the color, fill pattern, arc angle, radius and edge lines of the slice being drawn. The PIE function maintains (XLAST, YLAST) as the center of the PIE defined.

**PIEXY**
A utility command to return a location on the perimeter of the slice drawn. This is useful when positioning text around a pie (circle) on the page. The PIEXY command assumes that a pie slice has previously been defined (drawn). Unpredictable results may occur if a slice has not been drawn and the PIEXY function is invoked. The location is returned in the (XLAST, YLAST) coordinate pair.

**FRAME**
Allows you to specify a background color for the data area of the graphics page. This function works in addition to the CBACK= option. (See GLOBAL statements for details.)

**BAR**
(X, Y) defines the upper-right hand corner of a bar (rectangle) whose lower-left hand corner is (XLAST, YLAST). You may specify the color and fill pattern of the bar.

**OTHER VARIABLES**
The remaining variables in panel one (1) serve as additional information sources for the requested functions. Some of these variables assume new meanings for different functions, though the same variable name is used. We will call these dual variables. One such dual variable is STYLE. STYLE may represent a character font if text is to be produced (LABEL) or a pattern to be used in an area fill (POLY).

**DEMONSTRATION**
To illustrate use of the ANNOTATE facility, the following short sample and a set of more complicated examples is included at the end of this paper.

In this example, we want to
1) draw a box (rectangle) in the middle of the screen
2) put the word 'ANNOTATE' in the middle of the box.
DATA BOX; LENGTH FUNCTION COLOR & B.;

XSYS='3'; YSYS='3'; /* screen % */

COLOR='RED';
FUNCTION='MOVE'; X= 30; Y= 30; OUTPUT;
FUNCTION='DRAW'; X= 70; Y= 30; OUTPUT;
FUNCTION='DRAW'; X= 70; Y= 70; OUTPUT;
FUNCTION='DRAW'; X= 30; Y= 70; OUTPUT;
FUNCTION='DRAW'; X= 30; Y= 30; OUTPUT;

COLOR='YELLOW';
TEXT='ANNOTATE';
FUNCTION='LABEL';

RUN;
PROC PRINT;

PROC GSLIDE ANNOTATE=BOX;
RUN;

The above code produced the following output

(See top section Panel 1.)

LIMITATIONS

As of this writing, the following limitations exist in the ANNOTATE facility:

GMAP | No restrictions
GSLIDE | |
GCONTOUR | |
GPLOT | VREVERSE option and
formatted axes
are not supported
GCHART | under development;
supports screen and
window coordinates
C3D | under development;
supports screen and
window coordinates
CREPLAY | under development

SUMMARY

Although the ANNOTATE facility can perform a myriad of specialty plots, it should not be considered the only way a particular feature may be implemented. For now, it is a good alternative if you need special application graphics.

In closing, I wish to quote Fromme and Bury from a paper presented at SUGI '83 in New Orleans:

"... A graphics package designed for production of finished output from high-level commands can’t be expected to anticipate every feature visualized by a user for a particular graph. Neverthe-
less, small changes in the standard product would often make a highly acceptable graph out of an unusable one. ..."

It is that specific need which the SAS/GRAPH ANNOTATE facility addresses.

SAS and SAS/GRAPH are the registered trademarks of SAS Institute Inc., Cary, NC, USA.
**Sample 1**

This example will place a descriptive character string at the termination of each of four plotted curves.


```plaintext
DATA ECON;
LABEL COSTS = 'Cost' Q = 'Output (units per day)';
IF _N_ = 1 THEN TCPREV = 0;
RETURN;
INPUT Q TVC;
TFC = TOC + TVC;
ATC = TFC / Q;
AFC = TOC / Q;
IF _N_ = 3 THEN MC = TOC - TCPREV;
COSTS = MC;
TCPREV = TOC;
CARDS;
1 20
2 25
3 36
4 37
5 38
6 41
7 47
8 56
9 66
10 72
11 83
RUN;
```

Create the ANNOTATE dataset. X and Y values are taken from the ECON dataset. We want labels at the end of each curve, so use only the values for the highest 'Q'. This value was known in this example and was hardcoded. PROC MEANS, or any other statistic procedure may be employed equally as well.

```plaintext
DATA LABELS;
SET ECON;
IF Q = 11 /* Highest valued 'Q'
THEN DO;
XSYS = '2'; YSYS = '2'; /* The data systems must be set to data valued reference.
POSITION = 'C'; /* Position the text up and to the right of the original point.
COLOR = 'WHITE'; /* USE SOFTWARE TEXT TO LABEL POINTS
STYLE = 'DUPLEX';
/* Create an observation for each plotted curve on the page,
```

```
X = Q; Y = MC; TEXT = 'MC'; OUTPUT;
X = Q; Y = ATC; TEXT = 'ATC'; OUTPUT;
X = Q; Y = AVC; TEXT = 'AVC'; OUTPUT;
END;
```

**Cost of Production**

*from MILLER Economics Today, 4th Edition*
This example places city names onto a SAS/GMAP map.

DATA MAP:
SET MAPS; STATES;
IF STATE=12;
RUN;

DATA MAPCITY:
SET MAPCITY;
RUN;

PROC GPROJECT DATA=MAPCITY;
OUT "MAPCITY";
*/

DATA MAC CITY:
SET MAPCITY;
*/

TITLE "FLORIDA - CITIES";
FOOTNOTE "NEW FACILIT";
PATTERN COLOR=ORANGE V=EMBLEM R=DEEP;
RUN;

Florida Cities

221
### Panel 1.

<table>
<thead>
<tr>
<th>OBS</th>
<th>FUNCTION</th>
<th>COLOR</th>
<th>XSYS</th>
<th>YSYS</th>
<th>TEXT</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>COMMENT</td>
<td>3</td>
<td>3</td>
<td></td>
<td>Draw box in middle of screen</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>MOVE</td>
<td>RED</td>
<td>3</td>
<td>3</td>
<td></td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>DRAW</td>
<td>RED</td>
<td>3</td>
<td>3</td>
<td></td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>DRAW</td>
<td>RED</td>
<td>3</td>
<td>3</td>
<td></td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>DRAW</td>
<td>RED</td>
<td>3</td>
<td>3</td>
<td></td>
<td>30</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>DRAW</td>
<td>RED</td>
<td>3</td>
<td>3</td>
<td></td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>7</td>
<td>COMMENT</td>
<td>RED</td>
<td>3</td>
<td>3</td>
<td>Put text in middle of box</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>8</td>
<td>LABEL</td>
<td>YELLOW</td>
<td>3</td>
<td>3</td>
<td>Annotate</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>NAME OF VARIABLE</th>
<th>USE IN ANNOTATE</th>
<th>TYPE</th>
<th>LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>horizontal coordinate</td>
<td>numeric</td>
<td>8</td>
</tr>
<tr>
<td>Y</td>
<td>vertical coordinate</td>
<td>numeric</td>
<td>8</td>
</tr>
<tr>
<td>XC</td>
<td>horizontal coordinate</td>
<td>character</td>
<td>16</td>
</tr>
<tr>
<td>YC</td>
<td>vertical coordinate</td>
<td>character</td>
<td>16</td>
</tr>
<tr>
<td>XSYS</td>
<td>horizontal reference system</td>
<td>character</td>
<td>1</td>
</tr>
<tr>
<td>YSYS</td>
<td>vertical reference system</td>
<td>character</td>
<td>1</td>
</tr>
<tr>
<td>HSYS</td>
<td>sizing reference system</td>
<td>character</td>
<td>1</td>
</tr>
<tr>
<td>WHEN</td>
<td>phase (pass) of generation</td>
<td>character</td>
<td>1</td>
</tr>
<tr>
<td>POSITION</td>
<td>character origin control</td>
<td>character</td>
<td>1</td>
</tr>
<tr>
<td>FUNCTION</td>
<td>ANNOTATE function request</td>
<td>character</td>
<td>6</td>
</tr>
<tr>
<td>SIZE</td>
<td>height of character (.H=)</td>
<td>numeric</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>width of line drawn</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANGLE</td>
<td>text baseline angle (.A=)</td>
<td>numeric</td>
<td>8</td>
</tr>
<tr>
<td>ROTATE</td>
<td>text rotation angle (.R=)</td>
<td>numeric</td>
<td>8</td>
</tr>
<tr>
<td>STYLE</td>
<td>character font (.F=)</td>
<td>character</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>area fill pattern</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLOR</td>
<td>color to be used (.C=)</td>
<td>character</td>
<td>8</td>
</tr>
<tr>
<td>LINE</td>
<td>linetype (.L=)</td>
<td>numeric</td>
<td>8</td>
</tr>
<tr>
<td>TEXT</td>
<td>character text</td>
<td>character</td>
<td>up to 200</td>
</tr>
</tbody>
</table>

---

2/2
Panel 2.

SYSTEM 1
data percentage
SYSTEM 7
data relative percentage
SYSTEM 3
screen percentage
SYSTEM 9
screen relative percentage
SYSTEM 5
window percentage
SYSTEM B
window relative percentage
SYSTEM 2
data value
SYSTEM 8
data relative value
SYSTEM 4
screen value
SYSTEM A
screen relative value
SYSTEM 6
window value
SYSTEM C
window relative value