

Paper 085-31

Improving Your Graphics Using SAS/GRAPH® Annotate Facility

David Mink, Ovation Research Group, San Francisco, CA

David J. Pasta, Ovation Research Group, San Francisco, CA

ABSTRACT

Have you ever created a graph with SAS/GRAPH® and really liked it ... except for one little thing? The Annotate facility in SAS/GRAPH allows you to enhance or change features of your plot or chart. This presentation introduces Annotate and provides practical examples to illustrate how to improve your graphs. You do not need to be an expert at SAS/GRAPH to use Annotate. Even though Annotate is an extremely powerful tool for creating entirely custom figures, with a few guidelines (and the Annotate macros) it can be easy to use for simpler tasks. In this presentation, we will explain the principles behind Annotate and give specific examples of some common uses: adding a label to a single point on a graph, re-labeling an axis, adding a custom box of text, and applying "white out" to unwanted text. You will learn basic building blocks of Annotate that can be applied to many different graph enhancement situations.

INTRODUCTION

This paper is intended for SAS programmers with a working knowledge of base SAS® and at least some exposure to SAS/GRAPH. The concepts of the Annotate facility will be fully introduced, followed by some examples that even an Annotate veteran may find helpful.

The Annotate facility in SAS/GRAPH is a very useful tool for enhancing or changing features of your graphical output. Often when creating graphs using SAS, you find one little part that you wish you could change or one little item you wish you could add that would make your output perfect. Annotate gives you the flexibility to add any drawing object or text, in any format or color, to any place on your graphical output. This flexibility is what makes Annotate such an important and useful tool to learn for anyone planning to create graphics with SAS.

Using a data set provided in SAS/GRAPH documentation, this tutorial goes through some concrete examples of how to apply an Annotate solution to common graphing problems.

- Example 1 shows you how to place text next to a single point in a scatter plot and illustrates the importance of the co-ordinate systems.
- Example 2 addresses the problem of customizing a logarithmic axis.
- Example 3 illustrates some of the Annotate macros by showing you how to build a customized legend.
- Example 4 illustrates how to "white out" unwanted items in your graph output.

The examples make the concepts of the Annotate facility in SAS/GRAPH more concrete. Much of the material here is taken from the SAS/GRAPH Reference Manual.

INTRODUCTION TO THE SAS ANNOTATE FACILITY

In order to use the Annotate facility of SAS/GRAPH, the first step is to create an Annotate data set. The Annotate data set is a specific SAS data set with specific variables that can be manipulated in ordinary ways. Each observation in the data set is a drawing command defined by the values placed in each of the specific variables. The next step is to tell SAS to execute the Annotate commands by including the following option in your SAS/GRAPH code:

```
/ANNOTATE=<annotate file name>
```

SAS/GRAPH interprets and executes the drawing commands along with the graph and creates output that has both included. You can even use PROC GANNO to produce output entirely from Annotate with no graph. The concept is straightforward; now let us look more closely at some of the details.

ANNOTATE DATA SET VARIABLES

An Annotate data set must contain variables with predefined names. Other variables can be present in the data set, but they will be ignored by the Annotate facility. The Annotate variables tell Annotate what to do, how to do it, or where to do it. Below is a table defining some of the important variables that will be used in our examples. For a complete list of Annotate variables, refer to the SAS/GRAPH documentation.

TABLE 1. ANNOTATE DATA SET VARIABLES

VARIABLE	DESCRIPTION
FUNCTION	Specifies the Annotate drawing action. Table 2 below gives a list of important functions.
X	The numeric horizontal coordinate.
Y	The numeric vertical coordinate.
Z	For three-dimensional graphs, specifies the coordinate for the 3 rd dimension.
HSYS	The type of units for the size (height) variable.
XSYS	The coordinate system for the X variable.
YSYS	The coordinate system for the Y variable.
ZSYS	The coordinate system for the Z variable (for three-dimensional graphs).
ANGLE	Angle of text label or start angle for a pie slice.
COLOR	Color of graphics item.
LINE	Line type of graphics item.
POSITION	Placement/alignment of text.
ROTATE	Angle of individual characters in a text string or the sweep of a pie slice.
SIZE	Size of the graphics item. Specific to the function. For example, size is the height of the character for a label function.
STYLE	Font/pattern of a graphics item.
TEXT	Text to use in a label, symbol, or comment.
WHEN	Determines if Annotate command is executed (B)efore or (A)fter the graph.

FUNCTIONS

The Annotate data set FUNCTION variable tells SAS what to do. The other variables all modify, specify, or locate the function. Below is a table listing some of the more important functions that will be used in our examples. For a complete list of Annotate functions, refer to the SAS/GRAPH documentation.

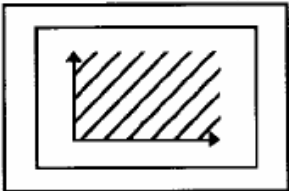
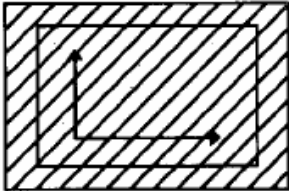
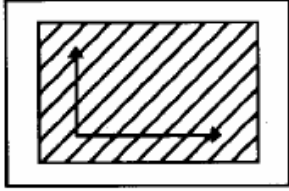
TABLE 2. FUNCTIONS

FUNCTION	DESCRIPTION
LABEL	Draws text.
MOVE	Moves to a specific point.
DRAW	Draws a line from the current position to a specified position.
COMMENT	As a documentation aid, allows you to insert a comment into the SAS Annotate file.
POLY	Specifies the starting point of a polygon.
POLYCONT	Continues drawing the polygon.
BAR	Draws a rectangle from the current position to a specified position
SYMBOL	Draws a symbol.
PIE	Draws a pie slice, circle or arc.

COORDINATE SYSTEMS

The Annotate facility recognizes three different drawing areas, each with two possible unit types and the ability to designate these unit types as relative or absolute. These 12 conditions are coded as 1-9, A, B, and C based on Figure 1 below.

FIGURE 1. AREAS AND THEIR COORDINATE SYSTEMS

	<u>Area</u>	<u>Unit</u>	<u>Coordinate System</u>	
	Data		Absolute	Relative
		%	1	7
	Graphics Output Area		Absolute	Relative
		%	3	9
	Procedure Output Area		Absolute	Relative
		%	5	B
		Values	2	8
		Cells	4	A
		Cells	6	C

The first of the three drawing areas is the Data Area which represents only the space within the graph axes. The second graph in the middle of figure 1 shows the Graphics Output Area which is the entire writable page of output. Finally, the bottom graph shows the Procedure Output Area or the area taken up by the graphic object. The Data Area can be referenced by percent or by the actual axis scale values. The Procedure Output and Graphics Output Areas can be referenced by percent or cell values. Each of these area-unit combinations can be referenced either absolutely in relation to the entire area or relatively in relation to the last drawn object. These three conditions are combined to determine the correct coordinate system code.

Once you have determined the appropriate coordinate system for each of your dimensions, you put the coded value in the XSYS and YSYS variables (and ZSYS for three-dimensional graphs). Based on the coordinate system(s), you chose where you want your graphics item to go, and assign those coordinate values to X and Y (and Z for three-dimensional graphs). X is the horizontal axis value and Y is the vertical axis value (Z is the depth value). This process gives you complete flexibility about where you put graphics items and how you reference those locations.

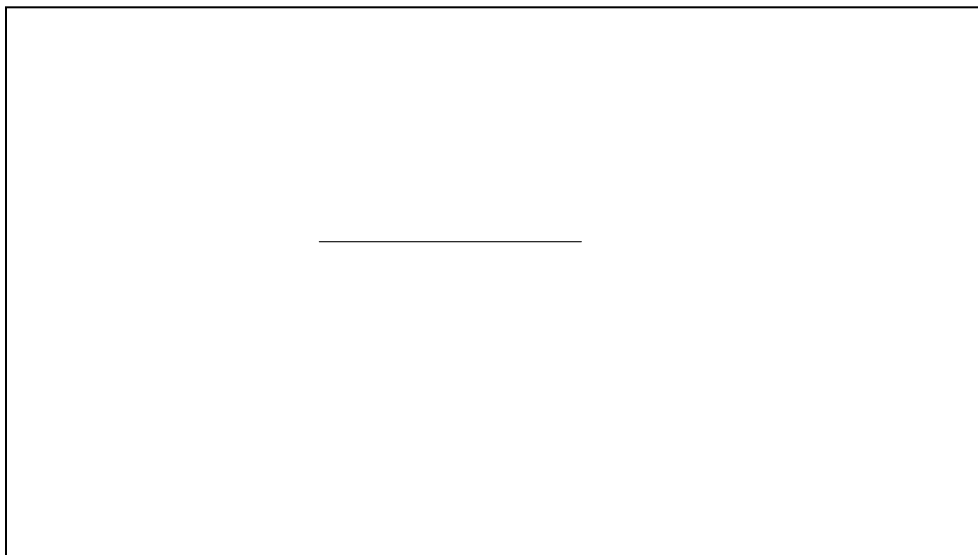
For example, say you wanted to place a label starting in the exact center of your page of output. You would select the Graphics Output Area and an absolute referenced percent unit. From figure 1 the code for this is '3'. The '3' is assigned to XSYS and YSYS so that when you give X and Y the value of 50 (for half way or 50 percent) the label begins there.

ANNOTATE MACROS

At this point it may seem as though creating SAS Annotate data sets is a tedious task. SAS has made it easier to accomplish by providing a set of Annotate macros. Consider the following data step that creates an Annotate data set to draw a black line:

```
data annotate_data_set1;
  length function style color $8;
  retain xsys ysys hsys '3';
  function='move'; x=35; y=25; output;
  function='draw'; color=black; line=1; size=1; x=65; y=25; output;
run;
```

FIGURE 2. A BLACK LINE



The Annotate data set created tells SAS to move to (35,25) and draw a black line to (65,25). This same object can be drawn more easily and more intuitively with the %line macro. Before any Annotate macros are used you must issue the %ANNOMAC macro call. This call tells SAS to include the Annotate macro library and makes the macros available for use. The following code illustrates how Annotate macros can perform the same task:

```
%annomac;
data annotate_data_set2;
  %dclanno;
  retain xsys ysys hsys '3';
  %line(35,25,65,25,black,1,1);
run;
```

The declarations have been simplified to one macro call (%dclanno) and the assignment and output statements in the first data step have been simplified into another macro call (%line). Table 3 lists some important Annotate macros that correspond to functions and their parameters.

Table 3. ANNOTATE MACROS

MACRO	DESCRIPTION
%DCLANNO	Declares the Annotate variables.
%LABEL(x, y, text-string, color, angle, rotate, size, style, position)	Places a label of text .
%MOVE(x, y)	Moves to a location.
%DRAW(x, y, color, line, size)	Draws a line from the current location to the specified location.
%COMMENT(text-string)	Allows an unexecuted comment to be inserted into the Annotate data set.
%POLY(x, y, color, style, line)	Begins drawing a polygon.
%POLYCONT(x, y, color)	Continues drawing a polygon.
%BAR(x1, y1, x2, y2, color, line, style)	Draws a bar.
%LINE(x1, y1, x2, y2, color, line, size)	Draws a line.
%PIEXY(angle, size)	Draws a pie slice.
%CIRCLE(x, y, size, color)	Draws a circle.

A LOOK AT OUR SAMPLE DATA

Now that you have an understanding of the Annotate facility, it is time to present some concrete examples. We will use the GIRIS2 sample data taken from the SAS/GRAPH manual. GIRIS2 contains Fisher's (1936) physical measurements of 150 irises. Fisher recorded the species (Setosa, Versicolor, and Virginica), petal length, petal width, sepal length, and sepal width of each iris. The examples here will focus on petal width and petal length for all species. Below is the first few records of this raw data.

TABLE 4. FISHER'S 1936 IRIS PETAL LENGTH AND WIDTH DATA

OBS	PETAL LENGTH IN mm	PETAL WIDTH IN mm	SPECIES
1	14	2	Setosa
2	56	22	Virginica
3	46	15	Versicolor
4	56	24	Virginica
...

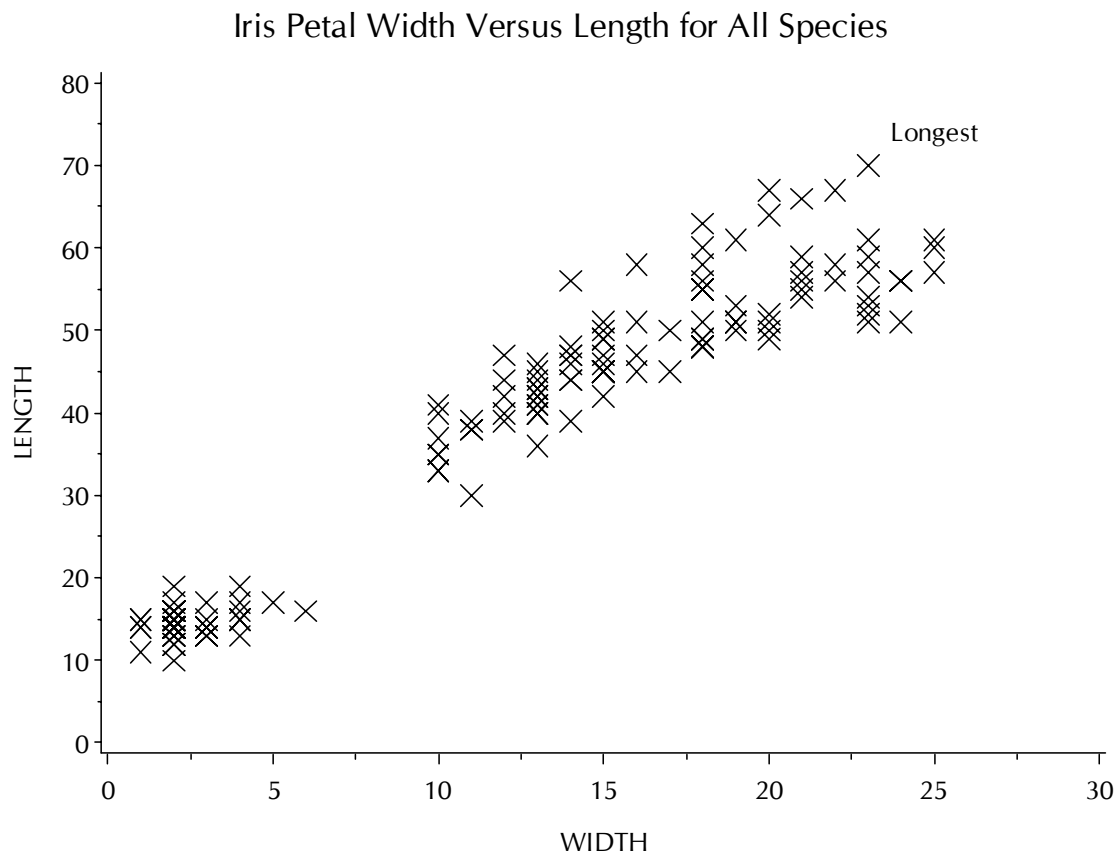
EXAMPLE 1: ADDING A LABEL TO A SINGLE POINT ON A GRAPH

The first example uses the Fisher iris data to create a scatter plot. Observation 90 has a length of 70mm and a width of 23mm. We would like to label this one point as "Longest". This can be done using Annotate. A standard GOPTIONS statement is assumed here and throughout the examples, but not shown. Consider the code to create the following plot.

```
proc gplot data=sample.giris2;
  title "Iris Petal Width Versus Length for All Species";
  axis1 order=0 to 80 by 10 minor=(number=1) label=(angle=90 "LENGTH");
  axis2 order=0 to 30 by 5 minor=(number=1) label=("WIDTH");
  symbol Interpol=none value="X" color=black height=2;
  plot petallen*petalwid / vaxis=axis1
                        haxis=axis2
                        noframe
                        annotate=annotate_data_ex1;

run;
quit;
```

FIGURE 3. ADDING A LABEL TO A SINGLE POINT ON A GRAPH



Notice the addition of the option,

```
annotate=annotate_data_ex1;
```

to the plot statement. This allows the plot to incorporate the Annotate instructions found in the Annotate data set. The following is the code that creates the Annotate data set called `annotate_data_ex1` and would have to precede the plot above:

```
data annotate_data_ex1;
  length function color $8;
  retain xsys ysys '2' hsys '3';
  function='label'; color='black'; x=23+2; y=70+5; text='Longest';
output;
run;
```

TABLE 5. ANNOTATE_DATA_EX1

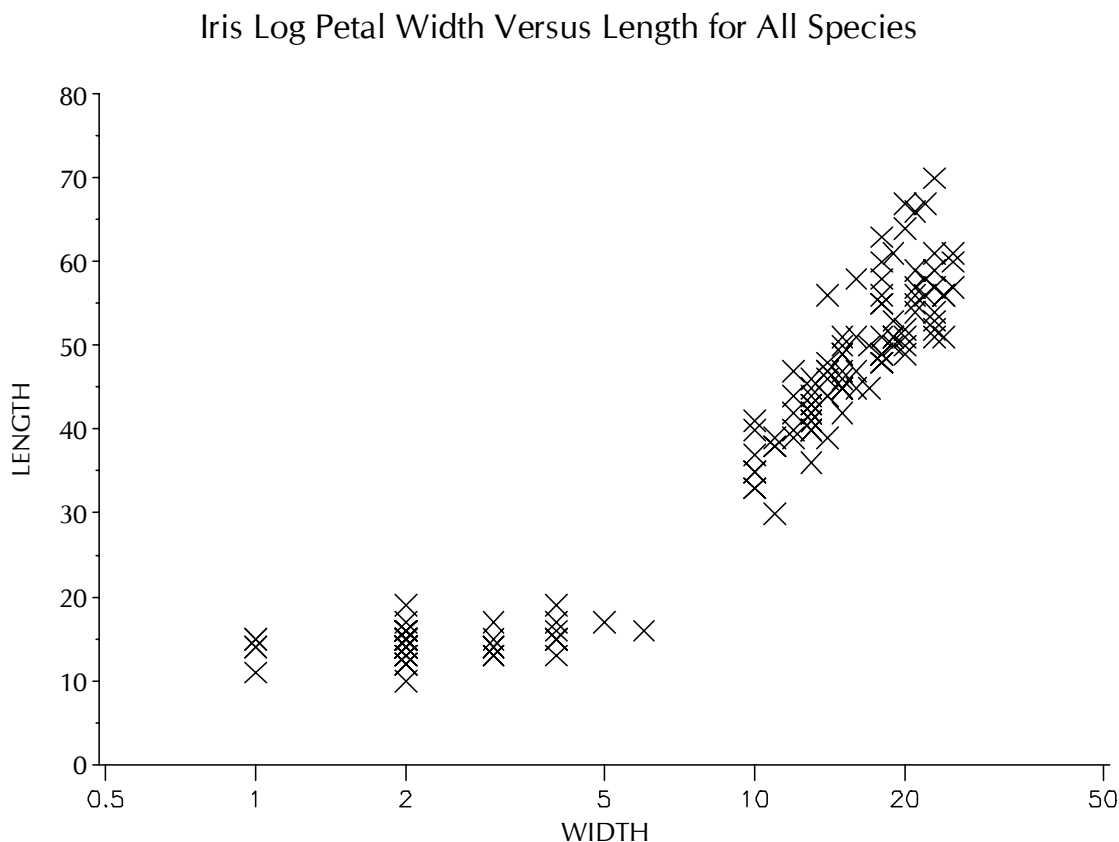
Obs	function	color	xsys	ysys	hsys	x	y	text
1	label	black	2	2	3	25	75	Longest

The coordinate system selected for the x axis and y axis is '2'. This corresponds (see Figure 1) to the Data Area with absolute values. With this coordinate system you can use the actual values of the point of interest to place the label. In Example 1, X is set to the X value plus 2 and Y is set to the Y value plus 5. This gives the label an offset so it is not written over the 'X'. Any of these points could be labeled in this manner and the power of Annotate should begin to emerge because you can see how simple it would be to assign X and Y values and perhaps even text based on variables in the data set itself.

EXAMPLE 2: RELABELING AN AXIS

Example 2 shows how you can customize an entire axis. The way SAS/GRAPH displays log-based scales is not as flexible as you may like. If you specify the individual major tick marks, SAS insists on spacing them equally. A solution is to add tick marks and labels using Annotate. In our example, we have converted the width axis to a log-based scale and then we use Annotate to specify tick marks and labels at 0.5, 1, 2, 5, 10, 20, and 50.

FIGURE 4. RELABELING AN AXIS



The Gplot SAS code is similar to Example 1 with the change in the definition of axis2, the horizontal width axis. The log base scale is defined and all tick marks and labels are removed.

```
proc gplot data=sample.giris2;
  title "Iris Log Petal Width Versus Length for All Species";
  axis1 order=0 to 80 by 10 offset=(0,0) minor=(number=1) label=(angle=90
"LENGTH");
  axis2 order=(0.5,50) minor=none major=none value=none logbase=10 label=(
" j=c "WIDTH");
  symbol Interpol=none value="X" color=black height=2;
  plot petallen*petalwid / vaxis=axis1
                        haxis=axis2
                        noframe
                        annotate=annotate_data_ex2;

run;
quit;
```

The Annotate data set must be defined to replace the tick marks and labels we left off the Gplot command. We know where on the x axis we want the tick marks so we'll start the short lines there. Then, using relative referencing in the graphical display area we can explicitly add the labels below each tick mark. Adding tick marks and labels in this manner gives you a great deal of flexibility when using logarithmic scales. Any labeling scheme that meets your needs can be easily implemented.

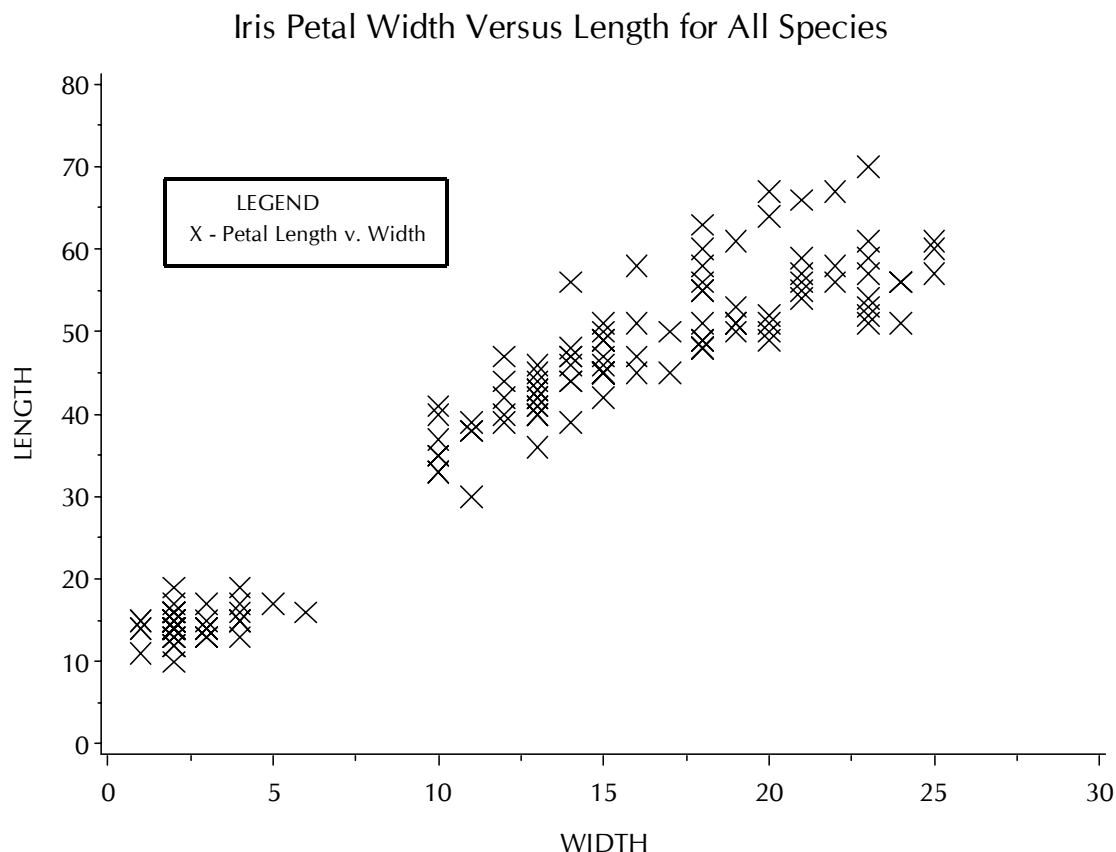
```
%annomac;
%macro tick(val);
  xsys='2'; ysys='2'; %move(&val.,0);
  xsys='B'; ysys='B'; %draw(0,-2,black,1,.25);
  %cntl2txt; %label(0,0,"&val.",black,0,0,3,simplex,E);
%mend tick;
data annotate_data_ex2;
  %dclanno;
  hsys='3';
  %tick(0.5); %tick(1); %tick(2); %tick(5);
  %tick(10); %tick(20); %tick(50);
run;
```

TABLE 6. ANNOTATE_DATA_EX2

bs	STYLE	FUNCTION	COLOR	XSYS	YSYS	HSYS	WHEN	POSITION	X	Y	LINE	SIZE	ANGLE	ROTATE	TEXT
1		MOVE		2	2	3	B	5	0.5	0	
2		DRAW	black	B	B	3	B	5	0.0	-2	1	0.25	.	.	
3		CNTL2TXT	black	B	B	3	B	5	.	.	1	0.25	.	.	
4	simplex	LABEL	black	B	B	3	B	E	0.0	0	1	3.00	0	0	0.5
5	simplex	MOVE	black	2	2	3	B	E	1.0	0	1	3.00	0	0	0.5
6	simplex	DRAW	black	B	B	3	B	E	0.0	-2	1	0.25	0	0	0.5
7	simplex	CNTL2TXT	black	B	B	3	B	E	.	.	1	0.25	0	0	0.5
8	simplex	LABEL	black	B	B	3	B	E	0.0	0	1	3.00	0	0	1
9	simplex	MOVE	black	2	2	3	B	E	2.0	0	1	3.00	0	0	1
10	simplex	DRAW	black	B	B	3	B	E	0.0	-2	1	0.25	0	0	1
11	simplex	CNTL2TXT	black	B	B	3	B	E	.	.	1	0.25	0	0	1
12	simplex	LABEL	black	B	B	3	B	E	0.0	0	1	3.00	0	0	2
13	simplex	MOVE	black	2	2	3	B	E	5.0	0	1	3.00	0	0	2
14	simplex	DRAW	black	B	B	3	B	E	0.0	-2	1	0.25	0	0	2
15	simplex	CNTL2TXT	black	B	B	3	B	E	.	.	1	0.25	0	0	2
16	simplex	LABEL	black	B	B	3	B	E	0.0	0	1	3.00	0	0	5
17	simplex	MOVE	black	2	2	3	B	E	10.0	0	1	3.00	0	0	5
18	simplex	DRAW	black	B	B	3	B	E	0.0	-2	1	0.25	0	0	5
19	simplex	CNTL2TXT	black	B	B	3	B	E	.	.	1	0.25	0	0	5
20	simplex	LABEL	black	B	B	3	B	E	0.0	0	1	3.00	0	0	10
21	simplex	MOVE	black	2	2	3	B	E	20.0	0	1	3.00	0	0	10
22	simplex	DRAW	black	B	B	3	B	E	0.0	-2	1	0.25	0	0	10
23	simplex	CNTL2TXT	black	B	B	3	B	E	.	.	1	0.25	0	0	10
24	simplex	LABEL	black	B	B	3	B	E	0.0	0	1	3.00	0	0	20
25	simplex	MOVE	black	2	2	3	B	E	50.0	0	1	3.00	0	0	20
26	simplex	DRAW	black	B	B	3	B	E	0.0	-2	1	0.25	0	0	20
27	simplex	CNTL2TXT	black	B	B	3	B	E	.	.	1	0.25	0	0	20
28	simplex	LABEL	black	B	B	3	B	E	0.0	0	1	3.00	0	0	50

EXAMPLE 3: ADDING A CUSTOM BOX OF TEXT

Example 3 constructs a legend from scratch using the Annotate facility. Although SAS can create a legend for you in a two-dimensional graph, you may want the flexibility to customize the look of a legend. G3D three-dimensional plots do not have a legend option so Annotate is your only choice for adding that type of information. The Fisher Iris scatter plot will be used again with the legend drawn in the upper left of the graphics area. This example also shows how using a relative reference co-ordinate system allows you to keep a group of annotate objects together to potentially move them around at once.

FIGURE 5. ADDING A CUSTOM BOX OF TEXT

The Gplot SAS code used to create the plot is the same as the code in Example 1, Fisher's iris petal length and width scatter plot. Please refer to example 1 for details; only the Annotate data set creation data step will be shown here.

```
%annomac;
data annotate_data_ex3;
  %dclanno; length text $30.;
  xsys='9'; ysys='9'; hsys='3';
  **Draw the initial legend box;
  %move(15,80);
  %draw(25,0,black,1,.5);
  %draw(0,-10,black,1,.5);
  %draw(-25,0,black,1,.5);
  %draw(0,10,black,1,.5);
  **Legend Text;
  %cntl2txt;
  function='LABEL'; color='black'; x=10; y=-2; text='LEGEND';
  size=3; output;
  function='LABEL'; color='black'; x=-1; y=-2; size=3;
  text='X - Petal Length v. Width'; output;
run;
```

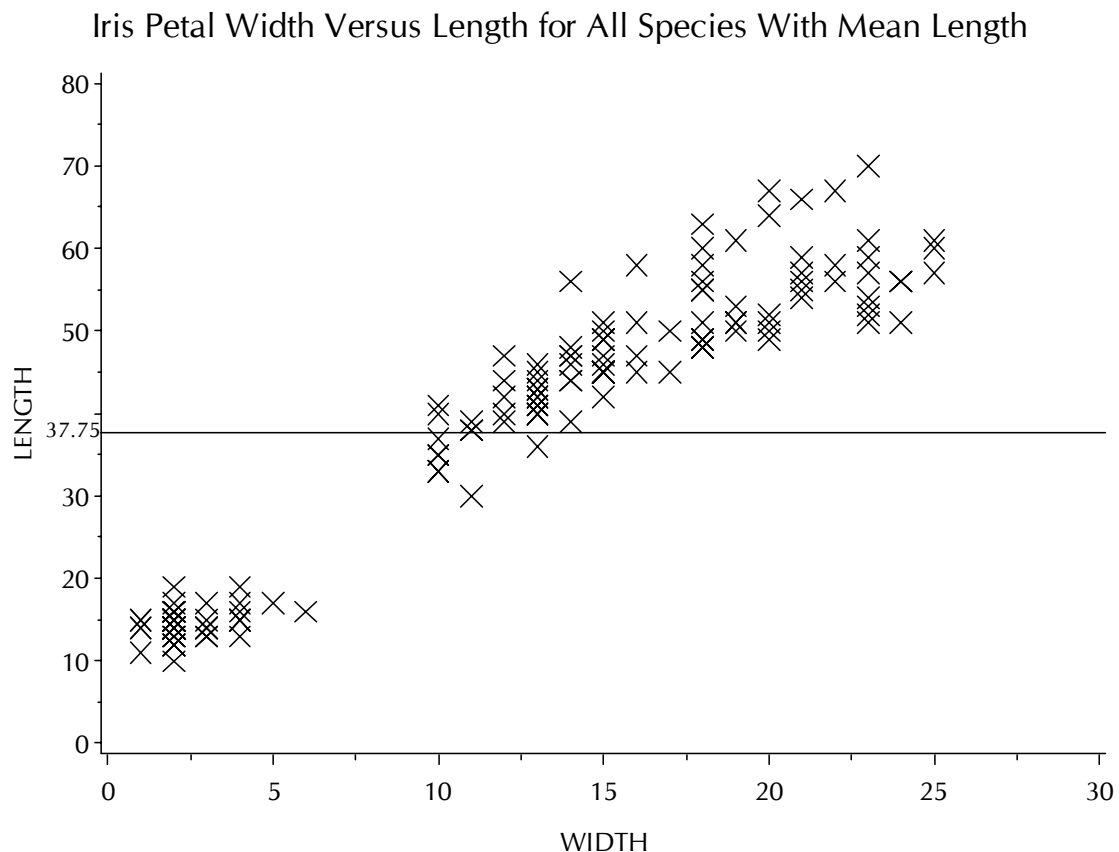

TABLE 7. ANNOTATE_DATA_EX3

Obs	STYLE	FUNCTION	COLOR	XSYS	YSYS	HSYS	WHEN	POSITION	X	Y	text	LINE	SIZE
1		MOVE		9	9	3	B	5	15	80		.	.
2		DRAW	black	9	9	3	B	5	25	0		1	0.5
3		DRAW	black	9	9	3	B	5	0	-10		1	0.5
4		DRAW	black	9	9	3	B	5	-25	0		1	0.5
5		DRAW	black	9	9	3	B	5	0	10		1	0.5
6		CNTL2TXT	black	9	9	3	B	5	.	.		1	0.5
7		LABEL	black	9	9	3	B	5	10	-2	LEGEND	1	3.0
8		LABEL	black	9	9	3	B	5	-1	-2	X - Petal Length v. Width	1	3.0

This simple legend is placed in the upper left hand corner of the graphics area. The rectangle is drawn with absolute coordinates and the text within the box is drawn relative to that box. Should you come back and want to move the legend only the rectangle coordinates need to be changed. The text referenced relatively will follow along. This allows you to create more complex objects as groups of Annotate basic objects and move them around as a group. You can see from this example the concept of creating a flexible customized legend. This is the only option for legends in G3D graphs.

EXAMPLE 4: "WHITING OUT" UNWANTED TEXT

The problem in example 4 is that of unwanted text and how to eliminate it. Suppose you want to add a vertical reference line to the Fisher Iris scatter plot representing the mean of all the lengths. In this data, the mean petal length is 37.75mm. This is a common requirement for many graphs, but can present a problem when labeling. We would like the reference line label to be along the y-axis with the tick mark labels, but it will conflict with the 40mm tick mark label. Since you want to keep the other labels and all the tick marks, the '40' is our only problem and must be eliminated. The Annotate facility allows you to create a rectangle the color of the background and the size of the unwanted label. Then a '37.75' can be added to label the reference line with no conflict.

FIGURE 6. "WHITING OUT" UNWANTED TEXT

The SAS/GRAPH code to create the above scatter plot is as follows. Please note the addition of the VREF option on the plot statement to generate the vertical reference line.

```
proc gplot data=sample.giris2;
  title "Iris Petal Width Versus Length for All Species With Mean Length";
  axis1 order=0 to 80 by 10 minor=(number=1) label=(angle=90 "LENGTH");
  axis2 order=0 to 30 by 5 minor=(number=1) label=("WIDTH");
  symbol Interpol=none value="X" color=black height=2;
  plot petallen*petalwid / vaxis=axis1
                        haxis=axis2
                        vref=37.75
                        noframe
                        annotate=annotate_data_ex4;

run;
quit;
```

The data step that creates annotate_data_ex4 should precede the above GPLOT. Using graphics output area coordinates, the '40' tick mark label is located. A background color box is drawn using the %bar macro. Then the example adds to the Annotate data set the label for the reference line.

```
%annomac;
data annotate_data_ex4;
  %dclanno;
  xsys='3'; ysys='3'; hsys='3';
  **White-Out Rectangle;
  when='A';
  %bar(5,55,8.5,51,white,3,solid);
  **Reference Line Label;
  function='label'; color="black"; x=7; y=51.5; text='37.75';
  size=3; style=""; output;

run;
```

TABLE 8. ANNOTATE_DATA_EX4

Obs	STYLE	FUNCTION	COLOR	XSYS	YSYS	HSYS	WHEN	POSITION	X	Y	LINE	SIZE	text
1		MOVE	white	3	3	3	A	5	5.0	55.0	.	.	
2	solid	BAR	white	3	3	3	A	5	8.5	51.0	3	0	
3		label	black	3	3	3	A	5	7.0	51.5	3	3	37.75

Using the "white out" concept above, you can eliminate a wide variety of unwanted graph marks and items.

CONCLUSION

What we have presented here is a brief introduction to the Annotate facility with examples. Those who are interested in learning more are directed to the SAS/GRAPH Manuals where additional topics on the use of Annotate in GCHART, GMAPS, and G3D graphics are presented. You can also learn to draw circles, pies, and arcs. If you are a very interested reader, you may want to also explore the subtleties of how SAS keeps track of the locations and positions of graphics items and character strings.

Annotate is a powerful flexible tool that will allow you to draw any text or object on any graphical output. Try these examples and others in the SAS/GRAPH manual and you will learn never to settle for a graph that is perfect...except for one little thing.

REFERENCES

SAS Institute Inc. 2004, *SAS/GRAPH® Software: Reference, Version 9*, Cary, NC: SAS Institute Inc.

CONTACT INFORMATION

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

David Mink
Ovation Research Group
120 Howard Street, Suite 600
San Francisco, CA 94105
(415) 371-2100
dmink@ovation.org

David Pasta
Ovation Research Group
120 Howard Street, Suite 600
San Francisco CA 94105
(415) 371-2100
dpasta@ovation.org

SAS and SAS/GRAPH are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

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