

## Paper 137-27

## Enhancements to SAS/GRAPH® in V9

Himesh Patel and Ravi Devarajan, SAS, Cary, NC

**ABSTRACT**

Would you like to improve the quality of your graphs and web-based reports using client technology? How about using annotate on your client output or, generating images in batch mode from clients? These are some of the enhancements made to SAS/GRAPH in order to improve the quality of output produced by ODS and the client side device drivers. V9 SAS/GRAPH also has new and improved GUI for web based ActiveX and Java graphs. Several improvements have also been made to the existing graphics procedures, including detailed options for the pie chart, and new area bar and radar charts for our Java and ActiveX based devices.

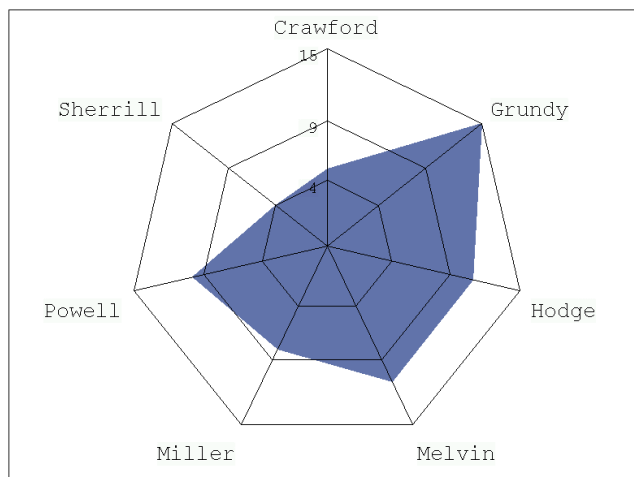
**INTRODUCTION**

This paper gives an overview of all the enhancements that have been made to SAS/GRAPH since V8.2. The additions to the traditional SAS/GRAPH procedures are described in SAS/GRAPH ENHANCEMENTS section of the paper. This is followed by sections that outline DRIVER and ODS ENHANCEMENTS to SAS/GRAPH in V9. The ACTIVEX GRAPH CONTROL and the JAVA GRAPH APPLET sections explain additions that enhance the client side capabilities of SAS/GRAPH. The SAS/GIS® ENHANCEMENTS section lists new features that have been added to SAS/GIS in this release. Finally, the NEW VISUALIZATIONS section describes new additions to the SAS/GRAPH family of products.

**SAS/GRAPH ENHANCEMENTS****GRADAR – SPIDER OPTION**

The SPIDER option draws a "spider web" on the radar chart connecting the spokes instead of the default tick marks on each individual spoke.

```
data state;
input name $ points rebounds blocks date date9.
color $;
cards;
Hodge      11   7   5 05Jan2002 red
Melvin     11   6   4 05Jan2002 blue
Powell     10   5   2 05Jan2002 green
Crawford   5    2   2 05Jan2002 black
Grundy     15   5   1 05Jan2002 pink
Miller     8    0   0 05Jan2002 gray
Sherrill   4    0   1 05Jan2002 purple
;
proc gradar data=state;
  chart name / freq =points
            starfill=(solid)
            cstarfill=(blue)
            spider;
run;
```



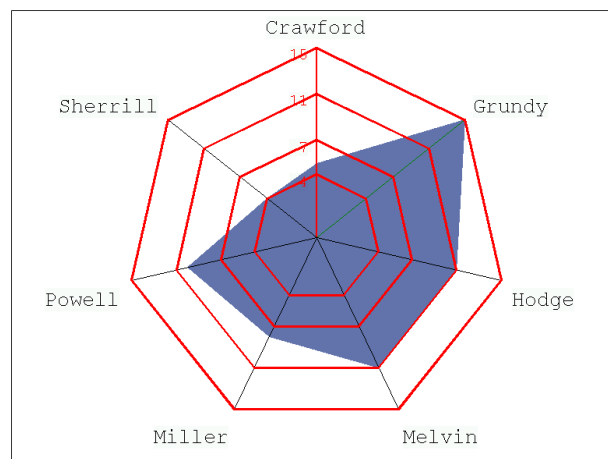
Output-1

If AXIS statements are specified with the SPIDER option, the tick mark options (NUMBER, WIDTH, and COLOR) specified for the first axis are then "transferred" to the values of the spider web lines. For example, if the first two AXIS statements specify:

```
Axis1 major=(number=4 color=red) width=2;
Axis2 color=green major=(number=6 color=cyan);
```

then the graph will contain four red spider web lines with a line width of 2 for the whole graph as shown in *Output-2*. Any options specified for the major tick mark on any other AXIS statements after the first one are ignored.

```
proc gradar data=state;
  chart name / freq =points
            staraxis = (axis1 axis2)
            starfill=(solid)
            cstarfill=(blue)
            spider;
```



Output-2

### LONG COLOR NAMES

In V9, color names greater than 8 characters long can be specified. Here is an example of such a name:

```
title1 color=CornFlowerBlue 'Proc Output';
```

Colors can also be defined through the SAS color registry. Color names containing spaces must be specified in quotes.

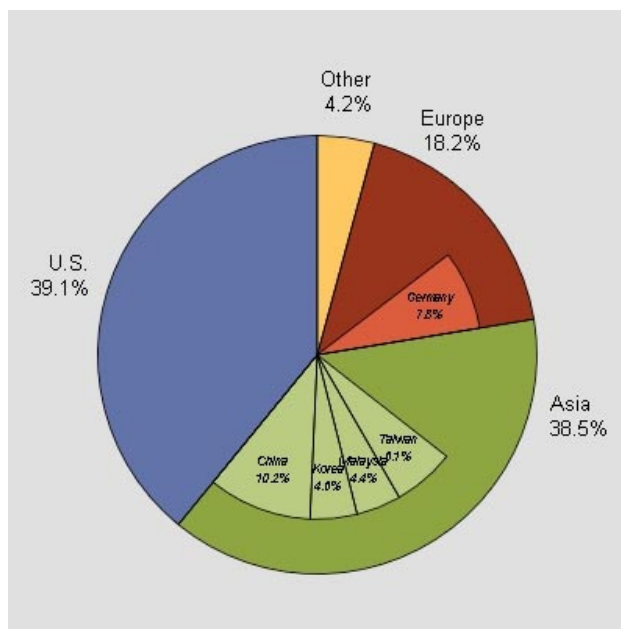
### DETAIL PIE

The `DETAIL=<variable>` option, on a `PIE` or `DONUT` statement, produces a pie or donut chart, with an inner pie overlay. This inner pie aligns with the outer pie's slices and displays the major components that make up the outer pie's slice in descending order.

*Output-3* shows an example of a detail pie looks like. To display the detail information within each country, you must specify `DETAIL=<region>`. In addition to `DETAIL` option, you can also specify `DETAIL_PERCENT=BEST` to display the percentage values using the best-fit algorithm. *Output-3* displays the countries that contributed greater than 4% of each region's value.

```
data countries;
  input country $ 1-14 region $16-26 Machinery;
  cards;
Taiwan      Asia      6.1
Korea       Asia      4.6
Malaysia    Asia      4.4
Malaysia2   Asia      3.9
Malaysia4   Asia      3.9
Malaysia5   Asia      1.5
U.S.        U.S.      39.1
Belgium     Europe    2.6
Germany     Europe    7.8
United Kingdom Europe    3.9
France      Europe    3.9
Santa       Antarctica 1.1
Bob         Antarctica 1.0
Cydonia     Mars      1.1
Tims House  Mars      1.0
China       Asia      10.2
Malaysia3   Asia      3.9
;
run;

proc gchart;
  pie region /
    angle=90
    detail_percent=best
    slice=outside
    percent=outside
    value=none
    sumvar=Machinery
    detail=country
    descending
  ;
run;
quit;
```



Output-3

In addition to the `DETAIL` and `DETAIL_PERCENT` options, you can also specify the following options to control the threshold within each detail slice and the radius of the detail pie.

`DETAIL_THRESHOLD=<0..100>`

indicates the percent threshold that determines if a detail slice is included in the inner pie. The default is 4%, which means only detail slices making up 4 or more percent of the whole pie will be displayed as a detail slice.

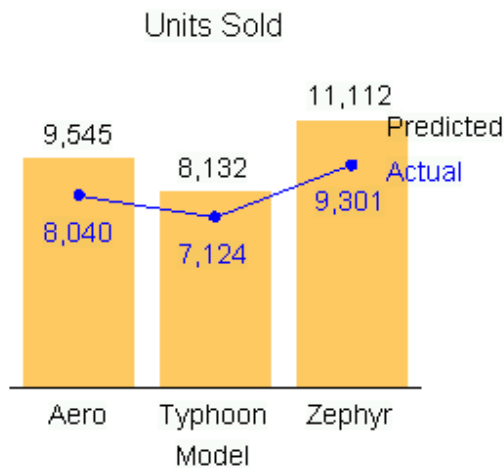
`DETAIL_RADIUS=<25..90>`

specifies the percent of the outer pie radius to use as the detail pie radius. The default is 75.

### ANNOTATIONS FOR THE CLIENT SIDE DEVICE DRIVERS

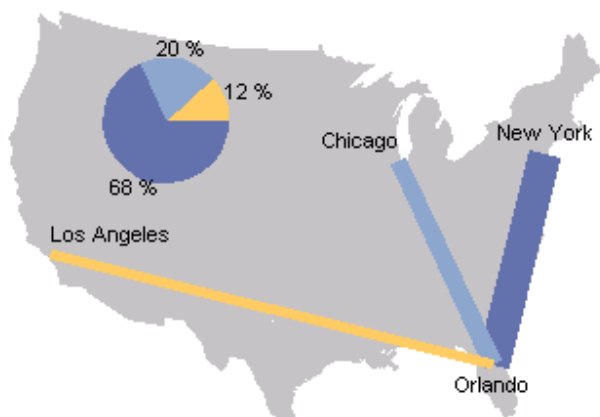
In V9, you can create Java or ActiveX-based graphs with user-defined annotation via the Output Delivery System. Support has been added for the `ANNOTATE=` option when used with the SAS/GRAPH procedures for Java and ActiveX series of device drivers. All annotate functions and variables are available in Version 9.0 with the exception of the `FRAME` and `IMAGE` functions. The syntax used for client annotation is the same as for SAS/GRAPH.

The chart in *Output-4* shows two response variables in a bar-line overlay chart and scatter point values with associated labels. The plot line, values and labels are done with annotation.



Output-4

Output-5 shows a map annotated with labels for cities, and lines of different styles connecting the cities and a Pie chart.



Output-5

**PROC MAPIMP (EXPERIMENTAL)**

The MAPIMP procedure is new experimental procedure that will import various map formats into SAS/GRAPH map data sets or into SAS/GIS data sets. This new procedure enables you to take advantage of the many sources of map data.

The initial release imports only Shapefiles from ESRI into SAS/GRAPH maps. Additional import formats will be added in future releases.

**DRIVER ENHANCEMENTS IN SAS/GRAPH**

The main improvement in driver support is that the PDF output via Universal Printing (GOPTIONS DEVICE=SASPRTC PRINTERPATH = PDF) is compressed in V9, which produces files 40 to 90 percent smaller. Most enhancements in Postscript, PCL, and PDF support are going to come via Universal Printing, which can be used with or without ODS.

Three new device drivers have been added:

- The JAVA2 device accesses more fonts using the Java plug-in.
- The ACTXIMG driver generates an image of the graph that would be rendered by a live ActiveX graph control.

- The JAVAIMG driver generates an image of the graph that would be rendered by a live Java graph applet.

**ODS ENHANCEMENTS IN SAS/GRAPH**

**CLIENT SUPPORT FOR ODS GRAPH STYLES**

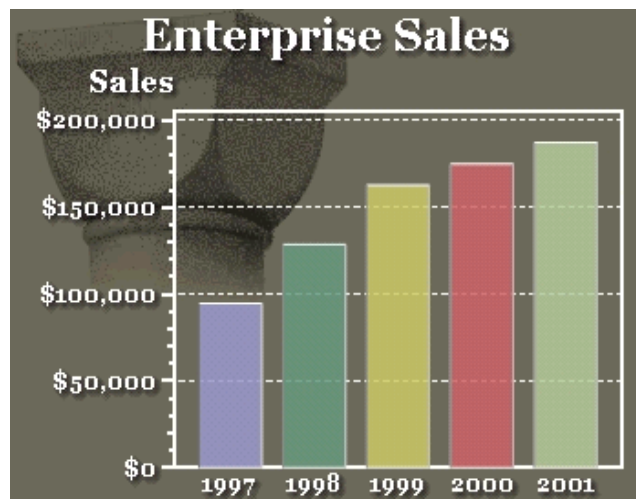
A graph style is a set of pre-defined parameters that affect the overall appearance of a graph. Styles cannot only make the graph output more visually pleasing, but also enhances its readability and consistency. They provide you with the ability to quickly and easily select from a variety of different visual styles.

Noteworthy additions to the rendering features of the ActiveX graph control and the Java graph applet make these styles available. New rendering capabilities incorporated for the styles include transparency effects, soft drop shadows, and enhanced font support. In V9, the graph styles are fully supported by the client side ActiveX and Java technologies. By simply changing the style in the graph properties dialog, the entire graph changes to that style. Different aspects of these styles such as, transparency and drop shadows can also be modified interactively.

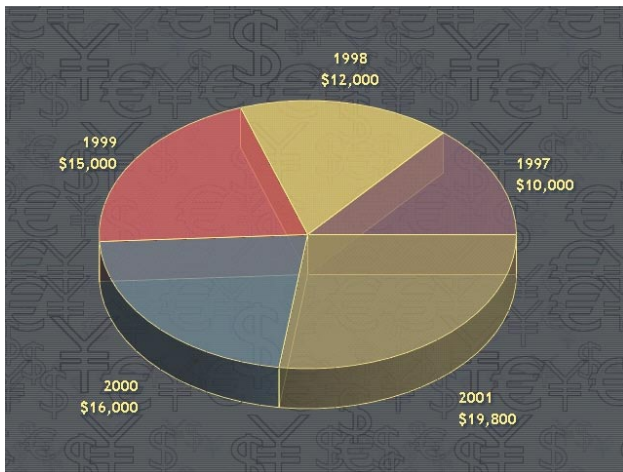
You can access the predefined styles as an ODS option. PROC TEMPLATE can be used to extend or modify the default style definitions.

Now in V9, ODS styles affect both the table and client graph output (the classic GRSEG output is not affected). The styles in ODS provide you with a mechanism for creating a visually appealing output, regardless of the procedure that is used. With the introduction of the graph styles, a consistent visual style is maintained for the entire ODS output, across tables and graphs. In addition, the previously available ODS styles will also be updated with color schemes that will make the graphs blend well with the established look of those styles. A total of sixteen predefined styles will be available with the ODS style option.

Output-6 shows one of the predefined styles called "Education" and Output-7 shows the "Money" style. Note that the image in the background and the reference line show through the bars and the chart area due to the effects of transparency.



Output-6



Output-7

### SERVER SIDE RENDERING OF GRAPH CLIENT OUTPUT

In V8.2, an experimental version of "server side rendering" for ODS PRINTER and ODS PDF was released, which allowed you to embed SAS/GRAPH client (ActiveX and Java) image output into ODS output. Starting with V9, you can generate the client images for HTML, RTF, PRINTER and PDF output from ODS. To use this functionality, you simply need to specify the ACTXIMG or JAVAIMG devices. For situations where you do not need live-client functionality in your output, this mechanism gives you an easy way to embed the new look of the client output in your reports.

### IMPROVED LIVE-CLIENT SUPPORT FOR ODS RTF

In V9, the ODS RTF output using a live ActiveX graph control is not only viewable from a local RTF file, but also from RTF files located on the web. This enhancement allows you to web publish Microsoft Word reports that use the client and view them anywhere on the network.

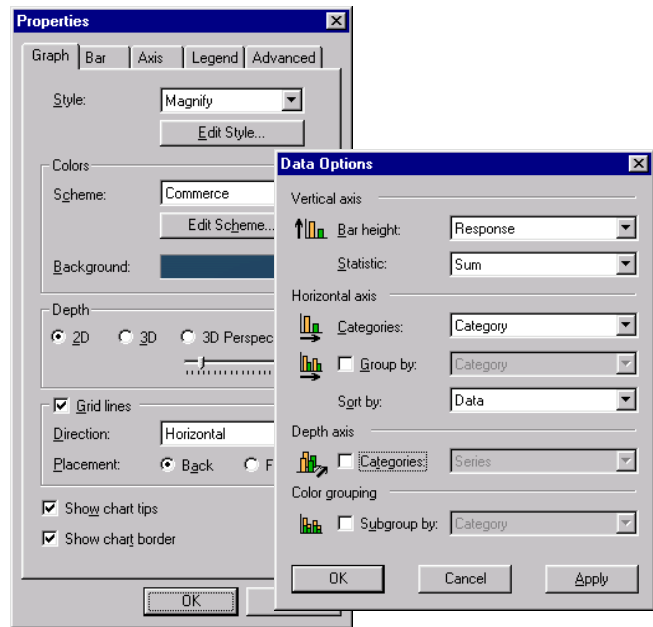
### GRAPH SUPPORT FOR ODS TAGSETS

In V8.2, ODS released an experimental version of tagset functionality that allowed you to customize the form and content of the table output. Now, with the production release of tagsets in V9, SAS/GRAPH output through ODS may also be modified through the use of client devices (JAVA or ACTIVEX).

### USER INTERFACE ENHANCEMENTS FOR CLIENT GRAPHS

The user interface is one of the most important parts of interactive graphs. Continued usability studies have resulted in several improvements over the V8.2 interface. The separate dialogs used in the V8.2 control have been combined into a single property sheet in V9, allowing users more efficient control in changing the graph properties associated with charts, axes, legends, graph styles, color schemes, and other advanced rendering options. The data options dialogs have also been redesigned to make interactive assignment of variables to chart roles more intuitive and to allow interactive selection of statistics and data sorting. *Output-6* shows one of the new ActiveX graph control property sheets and the DATA options dialogs.

To change Graph Styles or Color Schemes, select "Edit Style..." This opens a property sheet dialog that allows full control over setting of colors, fonts, textures, transparency, shadows, and other effects that make up the Graph Styles.



Output-8

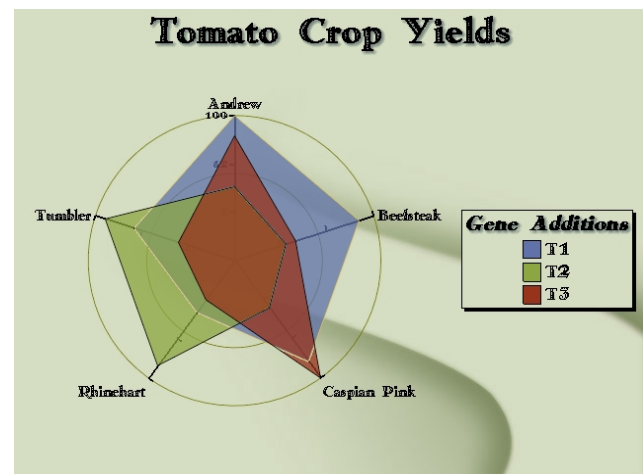
### ACTIVEX GRAPH CONTROL

#### RADAR CHART

The radar chart is based on PROC GRADAR. In addition to supporting most of the functionality of the radar chart, this version has improved appearance with the new styles that all of the ActiveX charts have. To generate an ActiveX radar chart, use ODS and set GOPTIONS device to ACTIVEX or ACTXIMG. An example of this chart is shown below in *Output-9*.

```
ods html file="C:\radar.html";
goptions dev=activex;
proc gradar data=tomatoes;
  chart variety / freq = tons
              overlay = gene;
run;
quit;

ods html close;
```



Output-9

### PROC GBARLIN (EXPERIMENTAL)

GBARLIN (Bar Line Overlay) is a new experimental procedure in

V9. It enables the creation of a chart consisting of ungrouped vertical bars with line plot overlays. This initial release only allows the creation of ODS output graphics for device of `ACTIVEX` or `ACTXIMG`. For example, the following code produces a bar line overlay plot:

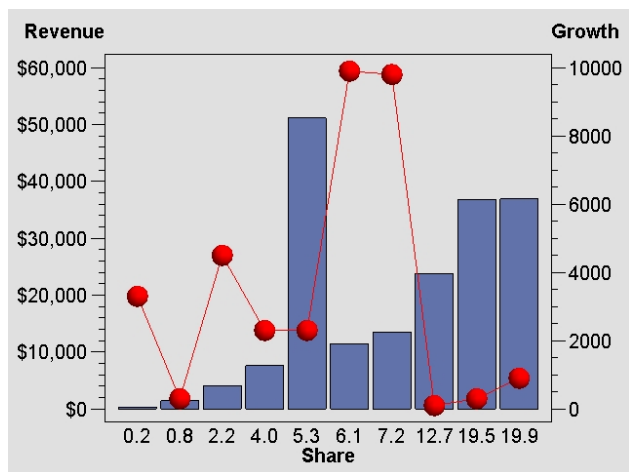
```
ods listing close;
ods html body="c:\temp\BarLineOverlay.html";
goptions device=activex;

data advertising;
format Revenue dollar12.;
Input Media $ 1-18 Year Revenue Share Growth;
datalines;
Newspapers      1985 25170 26.6  11
Magazines       1985 5155  5.4   6
Farm Publications 1985 186  0.2  11
Broadcast TV    1985 20298 21.4 22
Cable           1985 724  0.7   6
Radio           1985 6490 6.8   1
Radio           1997 13491 7.2   98
Direct Mail     1985 15500 16.4 34
Business Papers 1985 2375  2.5   4
Outdoor        1985 945  1.0  54
Yellow Pages    1985 5800  6.1   2
;
run;

title1 "Advertising Revenue Growth";

proc gbarlin data=advertising;
vbar share /
    sumvar=Revenue
    discrete
    plot=growth;
run;
quit;

ods html close;
ods listing;
```



Output-10

**AREA BAR CHART**

Area bar graphs are the only variation of bar graphs in which the width of the bars is significant. In the area bar graph, the widths of the bars are proportional to some measure or characteristic of the data element represented by the bars. In V9, PROC GAREABAR is only supported for `ACTIVEX` and `ACTXIMG` device drivers.

```
ods listing close;
ods html file='areabar.html';
goptions dev=activex;
data widget;
input product $ percent widgets office $;
cards;
Widget A      20    41    Raleigh
Widget A      10    53    Washington
Widget A      20    17    Buffalo
Widget A      40    12    Chicago
Widget A      10    11    Charleston
Widget B      15    55    Raleigh
Widget B      35    73    Washington
Widget B      10    29    Buffalo
Widget B      30    29    Chicago
Widget B      10    44    Charleston
Widget C      13    67    Raleigh
Widget C      27    95    Washington
Widget C      21    91    Buffalo
Widget C      19    51    Chicago
Widget C      20    30    Charleston
Widget D      15    71    Raleigh
Widget D      15    13    Washington
Widget D      15    72    Buffalo
Widget D      15    30    Chicago
Widget D      40    20    Charleston
Widget E      18    76    Raleigh
Widget E      12    19    Washington
Widget E      20    12    Buffalo
Widget E      20    28    Chicago
Widget E      30    65    Charleston
run;
```

```
PROC GAREABAR data=widget;
hbar product * widgets / sumvar=percent
subgroup=office;
run;
quit;
ods html close;
ods listing;
```

In *Output-11*, the percentage of widgets sold is the width variable and the response is the percent of a location's sales for that particular widget.



Output-11

## JAVA GRAPH APPLET

New graph styles are only supported in Java graph applet, but not in Map or Contour applet. In addition to the graph styles, several new options were added to make it more consistent with SAS/GRAPH output. Some of the major options to the applet include:

### BAR CHART

- Error bars (`ERROR` option) are now supported along with `CERROR` to set their color.
- Bars can be labeled with statistics using `OUTSIDE` and `INSIDE` options.
- `PATTERN` statements for images are supported for 2D bar charts.
- Bar charts support the `RANGE` option, which allows you to label bars by numerical range values.

### PIE CHART

- Pie charts with a group variable now support these options: `OTHER`, `HTML`, `INVISIBLE`, and `EXPLODE`.
- Interactive chart tips are now available for group pies.
- You can control the color of the 'other' slice with the `OTHERCOLOR` option.
- For `PATTERN` statements, `V=EMPTY` creates "hollow" pie slices.
- The `LABEL` option on Donut charts now supports font, height and color.

### PLOT CHART

New interpolation support:

- `BOX`
  - ◆ Options supported: `F`, `J`, `T`
- `STD`
  - ◆ Variance supported: `M`, `P`
  - ◆ Options supported: `B`, `J`, `T`, `BJ`, `TJ`
- `HILOC`
  - ◆ `HIGH`, `LOW` and `CLOSE` supported
  - ◆ Options supported: `B`, `J`, `T`, `BJ`, `TJ`
- Other `SYMBOL` statement options:
  - ◆ `COLOR=`, `CI=`, `CV=`, `CO=`
  - ◆ `FONT`, `HEIGHT` and `WIDTH`
  - ◆ `LINE=` (`JOINLINE` type only)
  - ◆ `POINTLABEL`, `VALUE`
  - ◆ `BWIDTH=` for box plots

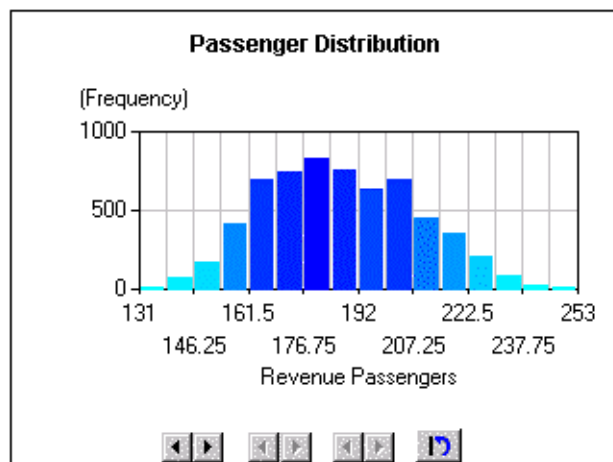
In V9, NLS enhancements to the GRAPH APPLET include Asian language support for menus in addition to the Danish, German, Spanish, French, Hungarian, Italian, Norwegian, Polish, and Swedish languages that were added in V8.2. Korean, Japanese, and Chinese are also supported.

## GRAPH-N-GO

In V9, two new graphical viewers have been added to GRAPH-N-GO along with several improvements to printing and exporting.

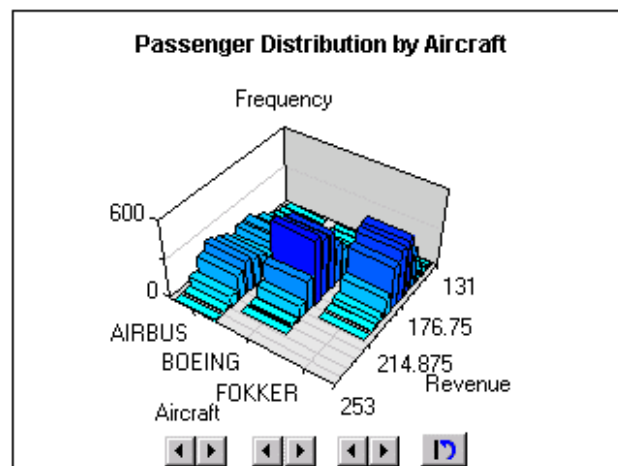
### HISTOGRAM

The HISTOGRAM object is a true 2D / 3D object that can show one or two categorical variables and a response variable in either a 2D or 3D view. `CATEGORY` variables can be either character or numeric. Numeric `CATEGORY` variables can be treated either discretely (BAR chart) or continuously (HISTOGRAM). For histograms you can choose the number of intervals (bins) for partitioning the range of values. *Output-11* shows a 2D view of a HISTOGRAM with 16 bins.



*Output-12*

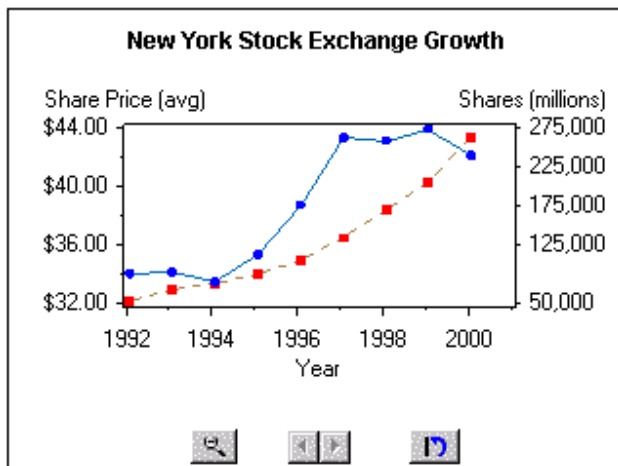
HISTOGRAM objects support a `SERIES` role that is a classifier on another axis. *Output-12* shows both `CATEGORY` and `SERIES` variables in a 3D PERSPECTIVE view. The controls in the viewer allow you to rotate, tilt and zoom the chart. These controls do not appear when you print or export the viewer.



*Output-13*

### OVERLAY PLOT

The OVERLAY plot object is a 2D object that supports the display of an X variable (numeric or character) and up to four Y numeric OVERLAY variables. You can choose to display any combination of the Y variables on either right or left vertical axes. You have control over the line styles, marker styles, and colors for each overlay.



Output-14

The controls within the viewer allow you to zoom in on the X-axis and scroll the plot left and right. This is useful when there are a large number of points.

Both the HISTOGRAM and OVERLAY plot viewers support SAS/GRAPH code generation as well as the other printing and exporting options.

#### OTHER NEW OR IMPROVED FEATURES

- Ability to use variable names or variable labels when setting up your data objects.
- Improved code generation using new V9 features.
- Export to HTML using ACTXIMG and JAVAIMG as well as GIF, ACTIVEEX, and JAVA drivers.
- Interactive tutorial for getting started.

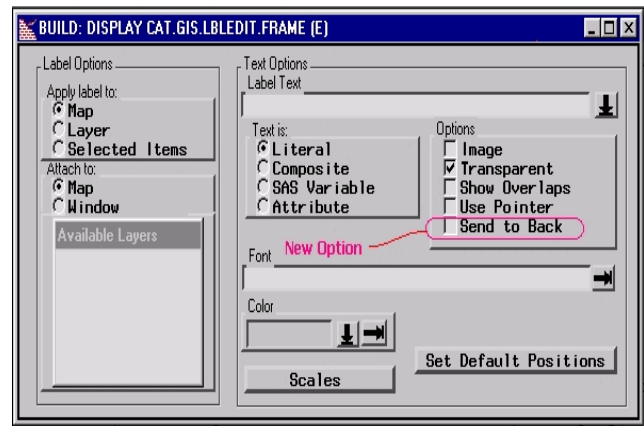
## SAS/GIS ENHANCEMENTS

### GIS DOCUMENTATION

The current manual *Using Spatial Data with SAS/GIS Software* was updated to reflect changes made for the 8.1, 8.2 and 9.0 releases. The book is now entitled *SAS/GIS 9: Spatial Data and Procedure Guide*. The web-based SAS/GIS tutorial and on-line help information have also been updated for V9.

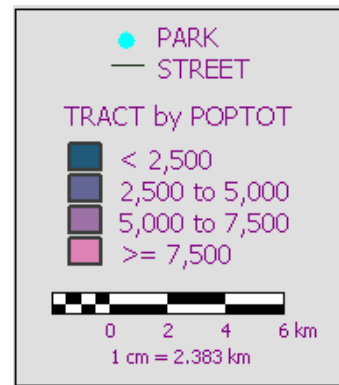
### GIS DESKTOP

All map labels are normally drawn over map features. However, this meant that large image labels obscured underlying map features. The ability to draw image labels beneath map features was added to 8.2, but there was no way to control this from the interactive GIS GUI. A new option checkbox 'Send to back' allows you to control whether image labels are displayed above or beneath map features.



Output-15

The graphical scale bar was redesigned to better fit within the map legend frame.



Output-16

### GIS PROCEDURE

The new CBACK option on the MAP statement allows you to set the map background color. This option is useful when you are creating maps in batch using ODS and displaying them on a web page.

```
MAP CREATE | REPLACE | UPDATE map-name /
    CBACK = colorName | CXrrggbb;
```

Spatial data can be merged to create larger map areas or to combine different types of map features. An EDGEMATCH merge matches common boundary chains between different regions. Missing values for left and right attribute variables along these common boundary chains are replaced with values from the adjoining area.

An EDGEMATCH merge is sometimes unable to merge all of the common chains in the spatial data. For example, two new options on the SPATIAL statement can alleviate this problem:

```
ERROR_ROW
```

prints detailed information about the left/right values of a bilateral variable for the specified chain data set row. It also prints the reason for the merge failure.

For example chains will fail to merge if zero is used instead of SAS missing values for a chain's left/right variable values.

```
ZEROMISSING
```

causes any zero in the chains' bilateral variables to be

treated as a missing value during the merge operation.

```
SPATIAL CREATE | REPLACE spatial-name /
  EDMATCH MERGE = (spatial-1, spatial-2, ...)
  ERROR_ROW = row
  ZEROMISSING;
```

The new **MAPLABEL** statement adds labels to the map and allows them to be modified in batch mode. This feature is very useful in creating complete maps with ODS for display on a web page. For example:

```
MAPLABEL CREATE | REPLACE | UPDATE | CONTENTS |
DELETE /
  TEXT = 'string'
  IMAGE = 'path' | fileref |
      lib.cat.entry.IMAGE
  ATTACH_TO = MAP | WINDOW
  ROW = integer
  MAP = lib.cat.map-entry
  DATASET = lib.dsname
  FORCE
  FRONT | BACK
  TRANSPARENT | NOTTRANSPARENT
  OVERLAPS
  FONT = fontname
  COLOR = color | CXrrggbb
  ONSCALE = ( label-on-scale
      real-units/map-units | METRIC |
ENGLISH )
  OFFSCALE = ( label-off-scale
      real-units/map-units |METRIC |
ENGLISH )
  POSITION = ( TOP | MIDDLE | BOTTOM
      LEFT | CENTER | RIGHT )
  OFFSET = ( x, y, REAL | PCT | PIXELS |
      x REAL | PCT | PIXELS,
      y REAL | PCT | PIXELS )
  ORIGIN = (x, y, REAL | PCT | PIXEL |
      x REAL | PCT | PIXELS,
      y REAL | PCT | PIXELS )
  _ALL_;
```

The new **LAYERLABEL** statement adds labels to specific map layers and allows them to be modified in batch. This is also very useful in creating complete maps with ODS for display on a web page.

```
LAYERLABEL CREATE | REPLACE | UPDATE | CONTENTS
| DELETE /
  TEXT = 'string'
  IMAGE = 'path' | fileref |
lib.cat.entry.IMAGE
  LAYER = lib.cat.layer-entry
  MAP = lib.cat.map-entry
  ROW = integer
  COMPOSITE = composite-name
  SAS_VARIABLE = variable-name
  ATTRIBUTE VARIABLE = link.variable-name
  DATASET = lib.dsname
  FORCE
  FRONT | BACK
  TRANSPARENT | NOTTRANSPARENT
  OVERLAPS
  POINTER | NOPOINTER
  FONT = fontname
  COLOR = color | CXrrggbb
  ONSCALE = ( label-on-scale
      real-units/map-units | METRIC |
ENGLISH )
  OFFSCALE = ( label-off-scale
      real-units/map-units |METRIC |
ENGLISH )
  OFFSET = ( x, y, PIXELS |
```

```
    x PIXELS, y PIXELS )
POSITION = ( TOPLEFT = integer,
  TOPCENTER = integer,
  TOPRIGHT = integer,
  MIDDLELEFT = integer,
  MIDDLERIGHT = integer,
  BOTTOMLEFT = integer,
  BOTTOMCENTER = integer,
  BOTTOMRIGHT = integer) )
  _ALL_;
```

## GEOCODING

Improved detection of input parameter errors and data problems has been implemented for the batch geocoding process.

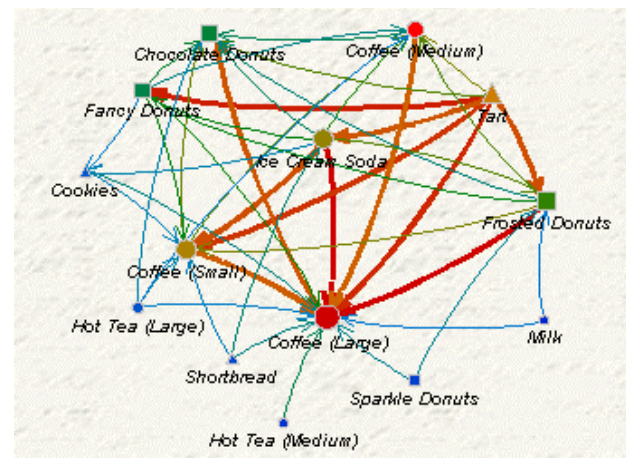
## NEW VISUALIZATIONS

### JAVA CONSTELLATION CHART APPLET

The Constellation chart applet displays a node link data with the links representing some relation among the nodes. The relationship could be sequential, affinity, or hierarchical. The applet uses the color, node shape, node size and link thickness to show the strength of these relations.

The **CONSTELLATION** chart applet can be used to display and analyze various kinds of node link data like web log, web click data, network flow data and simple affinity data. The nodes and links are displayed using force-directed layout as weighted directional graphs. Color, shape and size can be used to represent values of various relations among the nodes. *Output-17* shows a simple node-link diagram from a donut shop sale. Node shapes are used to group the type of item. The chart supports four layouts: Arcs, Association, Hierarchical and user specified.

The **Arcs** layout in *Output-17* displays all the nodes and links with the color, size, etc defined in the data. The user can interactively select a set of nodes of interest and see all the links coming to these nodes or going out of these nodes or just among these nodes.

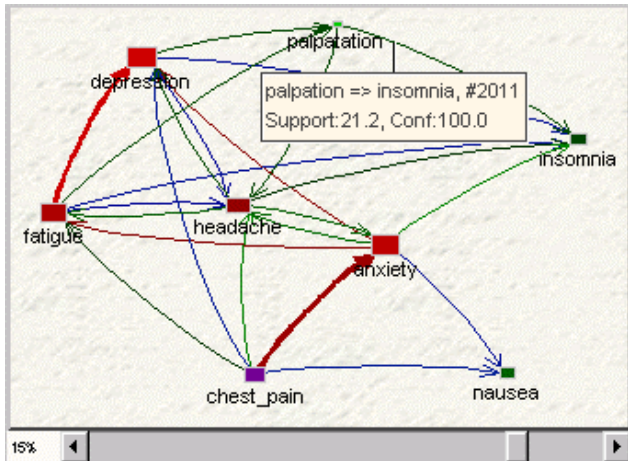


Output-17

In **Association** layout, color and link width may be mapped from the strength of the links. This also provides a slider, which can be used interactively to show only the links, which are stronger than the value of slider. *Output-18* takes diagnosis data from a medical file and displays it in the **Association** layout. The link between fatigue and depression seems the strongest in this data using the darkest color. Tool tips can store the detail information about each link and node. Each node can also store a URL link or

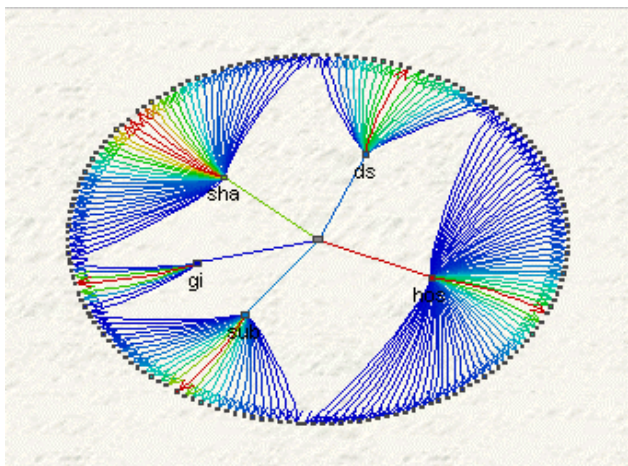


JavaScript command to provide more information interactively.



Output-18

Finally the Hierarchical layout in *Output-19* is used to visualize the data involving some hierarchies with multiple roots like file systems, web visitors etc. It finds all the roots in the data and connects them to a non-existing node to serve as the root of the tree. Data are then displayed as a tree. Each sub tree is a color ring. The red in middle represents the strongest link and then red to cyan to blue links fan out in descending strength.



Output-19

**XML INPUT**

Input data for the Constellation chart is provided in the form of XML script. The data can be in a separate XML file, or the XML script can be directly embedded in the HTML file. An XML code fragment is shown below.

```
<NodeLinkDiagram>
<Action>Home</Action>
<Nodes>
<Node>
<ID>0</ID>
<label>Coffee (Large)</label>
<shape>circle</shape>
<tip>Coffee (Large): Count 554, Beverages</tip>
</Node>
</Node>
```

```
<ID>1</ID>
<label>Tart</label>
<shape>triangle</shape>
<tip>Tart: Count 428, Desserts</tip>
</Node>...

</Nodes>
<Links>
<Link>
<fromNode>3</fromNode>
<toNode>0</toNode>
<value>106</value>
<tip>Ice Cream Sode=>Coffee; Count 106</tip>
</Link>
<Link>
<fromNode>4</fromNode>
<toNode>0</toNode>
<value>105</value>
<tip>Frosted Donuts=>Coffee; Count 105</tip>
</Link>...
</NodeLinkDiagram>
```

**DS2CONST MACRO**

A Constellation chart HTML generator macro, DS2CONST, is available in V9. This macro generates an HTML file with a scripted Constellation chart applet. Following is a simple DS2CONST macro that generates an HTML file.

```
data nodedata;
length nodeID value 8 label $11 tip $25;
input nodeID value @11 label $char11. @25 tip $char25.;
cards;
0 6556 depression depression: #6556
1 6322 anxiety anxiety: #6322
2 5980 fatigue fatigue: #5980
3 5286 headache headache: #5286
4 4621 chest_pain chest pain: #4621
5 3509 insomnia insomnia: #3509
6 3149 nausea nausea: #3149
7 2011 palpatation palpatation: #2011
;
run;

data linkdata;
length from to value 8 tip $40;
input from to value @13 tip $char40.;
cards;
2 0 5978 #5978, Support:63.0790, Conf:99.9833
4 1 4621 #4621, Support:48.7602, Conf:100.0000
1 0 4307 #4307, Support:45.4469, Conf:68.1272
1 2 3964 #3964, Support:41.8276, Conf:62.7017
2 3 3010 #3010, Support:31.7611, Conf:50.3429
0 3 3009 #3009, Support:31.7506, Conf:47.5957
1 6 2772 #2772, Support:29.2498, Conf:43.8469
0 5 2767 #2767, Support:29.1970, Conf:43.7678
2 5 2767 #2767, Support:29.1970, Conf:46.2786
4 6 2609 #2609, Support:27.5298, Conf:56.4596
4 0 2606 #2606, Support:27.4982, Conf:56.3947
3 5 2347 #2347, Support:24.7652, Conf:48.2227
4 2 2263 #2263, Support:23.8789, Conf:48.9721
7 3 2011 #2011, Support:21.2198, Conf:100.0000
2 7 2011 #2011, Support:21.2198, Conf:33.6344
7 2011 #2011, Support:21.2198, Conf:31.8096
.....more data;
title1 'Diagnosis Sequence Diagram.';

%ds2const (ndata=nodedata,
lndata=linkdata,
```

```
datatype=assoc,  
minlnkwt=30,  
height=650,  
width=600,  
codebase=&codebase,  
archive=constapp.jar,  
htmlfile=&htmlroot.diagnose.html,  
colormap=y,  
nid=nodeID,  
nlabel=label,  
nvalue=value,  
ntip=tip,  
lfrom=from,  
lto=to,  
lvalue=value,  
ltip=tip,  
linktype=arrow);
```

## CONCLUSION

In V9, major emphasis has been put into the appearance of the charts and the user interface of the web based output. With the new predefined styles, you can generate consistent looking graphs and tables in web-based reports through ODS. New charts and annotate support for the client side Graph applet and ActiveX graph control should allow you to generate and enhance a variety of different charts for your web-based applications. In addition to the procedure-based charts through ODS, V9 also offers Java applets that are driven through predefined macros.

## CONTACT INFORMATION

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

Himesh Patel ([Himesh.Patel@sas.com](mailto:Himesh.Patel@sas.com))  
Ravi Devarajan ([Ravi.Devarajan@sas.com](mailto:Ravi.Devarajan@sas.com))  
SAS Campus Drive  
Cary, North Carolina 27513

SAS and all other SAS Institute Inc. products 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 registered trademarks or trademarks of their respective companies.