

WEB ENABLE YOUR GRAPHS AND REPORTS

Jessica Bonow, Arkansas Foundation for Medical Care, Little Rock, AR

ABSTRACT

Arkansas Foundation for Medical Care (AFMC) serves to promote health and education activities to improve the health care in the state of Arkansas. The data driven structured approach to measure clinical improvement is evident in many of AFMC's projects.

Annually, AFMC applies HEDIS measure to calculate the Childhood Immunization Status for the Medicaid population in Arkansas. Utilizing PROC GCHART and Base SAS, a graphical representation has been created to illustrate the improvement of immunization rates over a selected time period. Using Version 8.2 Output Delivery System (ODS) to generate HTML and graphical output for the Web allows these results to be viewed by Internet users. Other procedures, options, and technologies covered in this paper include PROC REPORT, the drilldown option, and JAVA Applets. All in all, this paper gives an example of how one can attain the graphics and reports and use some of the options available with the JAVA device for Web graphics.

INTRODUCTION

Arkansas Foundation for Medical Care (AFMC), Arkansas' Quality Improvement Organization, seeks to promote excellence in health care through education and evaluation. AFMC has several contracts to review and to help improve the care furnished to Arkansas recipients. For the Medicaid contract, AFMC annually reviews the Childhood Immunization Status for the Medicaid population. The measures calculated were according to the Health Plan Employer Data & Information Set (HEDIS). The results of this review were represented graphically and in reports, by using PROC GCHART and PROC REPORT, respectively, to Medicaid providers and are soon to be available to all users of the Internet.

BASIC PROC REPORT INFORMATION

Base SAS has a flexible report-writing tool called PROC REPORT. With this procedure, you can control the layout, customize column headers, store and retrieve report definitions, report statements and your report profile, and control the windows environment. To begin the PROC REPORT procedure, the syntax is as follows:

```
PROC REPORT <option(s)>;
  BREAK location break-variable </ option(s)>;
  BY <descending> variable-1
    <...<descending>variable-n><notsorted>;
  COLUMN column-specification(s);
  COMPUTE location <target></STYLE=<style-
    element-name><[style-attribute-
    specification(s)]>>;
  LINE specification(s);
  ...select SAS language elements...
ENDCOMP;

COMPUTE report-item </type-specification>;
CALL DEFINE (column-id, 'attribute-
  name',value);
  ... select SAS language elements...
ENDCOMP;

DEFINE report-item /<usage> <attribute(s)>
  <option(s)> <justification>
  <COLOR=color>
  <'column-header-1'<... 'column-
  header-n'>><style>;

FREQ variable;
RBREAK location </option(s)>;
WEIGHT variable;
```

Many options exist for the PROC REPORT statement that allow you to control and customize your layout. I will give you details on the options I used in my SAS program. The first thing I did was specify an input data set using the DATA= option for all four reports. To customize the column headers by underlining all the column headers and the spaces between them at the top of each page of the report, the HEADLINE option was used. The environment chosen was a non-windows environment. When you use this type of environment, PROC REPORT runs without the REPORT window and sends its output to the SAS procedure output. The final option used for the PROC REPORT statement was the STYLE= option. It specifies one or more style elements for the Output Delivery System (ODS) to use for different parts of the report. The location for the STYLE= option is the header. Within the header, I am going to set the background, font_face, and font_weight attributes.

The COLUMN statement describes the arrangement of all the columns and of headers that span more than one column. This statement is where you determine

which report-items you want in the report. The column specifications are report-items, such as the name of a data set variable, a computed variable, or a statistic.

The COMPUTE statement starts a compute block, which contains one or more programming statements that PROC REPORT executes as it builds the report. In my reports, I create a compute block that displays a preimage of the AFMC logo near the top left hand corner of each page. Also, I changed the style for the text that is created by any LINE statements. The style is `font_weight=bold`, `font_style=italic`, `font_size=3`, and `font_face=times`. LINE statements are just subsets of the features of the PUT statement for writing customized summaries. I am creating a title for the reports with the LINE statements. Several specifications used were `@column-number`, which specifies the number of the column in which to begin displaying the next item, and `'character-string'`, which is the text to display as the title of the report. A line directly from the program is `line@20 "For State Fiscal Year 2001";`. This line puts at column number 20 the phrase, For State Fiscal Year 2001, in bold, italic, times, size 3 font. The ENDCOMP statement marks the end of a compute block. All compute blocks must start with a COMPUTE statement and end with an ENDCOMP statement.

When defining the report variables, all the variables must be assigned a usage type. This step requires the DEFINE statement. Six different types of usage are: GROUP, ORDER, DISPLAY, ANALYSIS, COMPUTED, and ACROSS.

The DISPLAY type lists all the values as they appear in the data set. DISPLAY is the default for character variables if no ORDER or GROUP variables exist. GROUP applies to class variables. The report can be separated as for each of the GROUP variables and the summary statistics will be calculated for some if not all of the other variables. For example, the sum can be calculated for all ANALYSIS variables for each different value of GROUP variable. ORDER sorts the data and creates groups for summary statistics. ANALYSIS is the default for numeric variables. Summary statistics such as SUM and MEAN can be easily calculated and displayed for the entire data set or each group or subgroup. ACROSS is a class variable that is used in the column dimension. COMPUTED variables are variables that are not input variables of the data set. These are variables that the user defines for the report. Additional information on the variable usage types can be found in the *SAS Guide to the Report Procedure*.

Specifying style attributes for a report item, options for a report item, controlling the placement of values and column headers, and specifying a style element for ODS for the report item are option categories for

the DEFINE statement. In my program, I used a limited amount of DEFINE statement options. I assigned a SAS or user-defined format to the item by the `FORMAT=` option. I defined the width of the column in which PROC REPORT displays the report item. Using `STYLE=` option, I changed the background color for one of the columns in the report.

After you have written the SAS code for creating the reports needed, making a report an HTML file and accessible to the Web is easy. By just adding two ODS statements and turning off the WINDOWS option, I can apply the reports to my drill-down bar charts and make it accessible to the users of the Internet.

PROC GCHART INFORMATION

When graphically displaying data or results, SAS users have numerous options. In this paper, I will be giving you information about the graphics procedure PROC GCHART. This procedure can produce six different types of charts: block charts, horizontal and vertical bar charts, pie and donut charts, and star charts. These charts will represent graphically the value of the statistic calculated for one or more variables in the input SAS data set. The statistics that are available are frequency or cumulative frequency counts, percentages or cumulative percentages, sums, and means. The procedure syntax is:

```
PROC GCHART<DATA=input-data-set>
```

```
<ANNOTATE=Annotate-data-set>
```

```
<GOUT=<libref>output-catalog>
```

```
<IMAGEMAP=output-data-set>
```

```
BLOCK chart-variable(s) </ options(s);>
```

```
HBAR | HBAR3D | VBAR | VBAR3D chart-  
variable(s) </ option(s);>
```

```
PIE | PIE3D | DONUT chart-variable(s) </  
option(s);>
```

```
STAR chart-variable(s) </ option(s);>
```

My focus is going to be on the vertical bar chart in three dimensions, VBAR3D. I will give you some of the options that I used in the program for the Childhood Immunization Status. All of these options will affect all graphs produced by the procedure. The major statement option that I used was the `DATA=input-data-set`. This option specifies the SAS data set that contains the variable(s) to chart. Once you have decided which type of chart you want to create, there are other options that change the look of the chart. All options can be divided into categories according to their functions. For the VBAR3D chart, I used the following options from the appropriate categories.

The appearance options used:

LEGEND=legend<1...99>

Assigns the specified LEGEND definition to the legend generated by the SUBGROUP=option. The LEGEND= option itself does not generate a legend. LEGEND= is ignored if SUBGROUP= is not used, the NOLEGEND option is used, or the PATTERNID=option is set to any value other than SUBGROUP; that is the value of PATTERNID= is BY or GROUP or MIDPOINT.

PATTERNID=by|group|midpoint|subgroup

Specifies the way fill patterns are assigned. By default, PATTERNID=SUBGROUP. Values for PATTERNID= are as follows:

BY changes patterns each time the value of the BY variable changes. All bars use the same pattern if the GCHART procedure does not include a BY statement.

GROUP changes patterns every time the value of the GROUP variable changes. All bars in each group is the same pattern, but a different pattern is used for each group.

MIDPOINT changes patterns every time the value of the MIDPOINT value changes. If you use the GROUP= option, the respective midpoint patterns are repeated for each group.

SUBGROUP changes patterns every time the value of the SUBGROUP variable changes. The bars must be subdivided by the SUBGROUP= option for the SUBGROUP value to have an effect. Without SUBGROUP=, all bars have the same pattern.

The statistic option that I used for the bar charts was SUMVAR=summary-variable, which specifies a numeric variable for sum or mean calculations. The GCHART procedure calculates the sum or, if requested, the mean of summary variable for each midpoint. The length of the bars along the response axis represents the resulting statistics, and they are displayed at major tick marks.

The midpoint options used are:

Discrete

Treats a numeric chart variable as a discrete variable rather than as a continuous variable. The GCHART procedure creates a separate midpoint and, hence, a separate bar for each unique value of the chart variable. If the chart variable has a format associated with it, each formatted value is treated as a midpoint.

GROUP=group-variable

Organizes the data according to values of group-variable. Group-variable can be either character or numeric and is always treated as a

discrete variable. GROUP= produces a separate group of bars for each unique value of the group variable. Missing values for group-variable are treated as a valid group. The groups are arranged in ascending order of the group variable values.

MIDPOINTS=value-list

Specifies the midpoint values for the bars. The way you specify value-list depends on the type of the chart variable. For numeric chart variables, value-list is either an explicit list of values, or a starting and an ending value with an interval increment, or a combination of both forms. For character variables, value-list is a list of unique character values enclosed in quotation marks and separated by blanks.

SUBGROUP=subgroup-variable

Divides the bars into segments according to the values of the subgroup-variable. A subgroup-variable can be either character or numeric and is always treated as a discrete variable. SUBGROUP= creates a separate segment within each bar for every unique value of the subgroup variable for that midpoint.

Used axes options are:

AUTOREF

Draws a reference line at each major tick mark on the response axis. To draw reference lines at specific points on the response axis, use REF= option. By default, reference lines in 3D are drawn on the back plane of the axis.

GAXIS=AXIS<1...99>

Assigns the specified AXIS definition to the group axis. A group axis is created when you use the GROUP= option. You can use the AXIS definition to modify the order of the groups, the text of the labels, and the appearance of the axis. GAXIS= is ignored if the specified AXIS definition does not exist.

MAXIS=AXIS<1...99>

Assigns the specified AXIS definition to the midpoint axis. The MAXIS= option is ignored if the specified AXIS definition does not exist.

RAXIS=AXIS<1...99>

Specifies values for the major tick mark divisions on the response axis or assigns the specified AXIS definition to the axis. By default, the GCHART procedure scales the response axis automatically and provides an appropriate number of tick marks.

Several global statements are allowed by the GCHART procedure. They include axis, legend, pattern, title, and footnote. This procedure supports the chart to have drill-down functionality.

WEB ENABLE USING ODS

To web enable graphs or reports is not difficult if using Version 8. By using the Output Delivery System (ODS) that is available to SAS users and by establishing and setting a few options, you could have your HTML documents up and running in no time. The ODS has many advantages over other methods of producing Web based output. With ODS, one can name the body file(s) for storing the ODS output, determine whether the footnotes and titles are written as part of the graph or part of the HTML document, combine graphics and non-graphics output in the web pages created, generate Table of Contents and Table of Pages to link to the output, create drilldown graphs, and have the ability to create frames to display information. A minimal amount of steps are needed to use the ODS with SAS/GRAPH. The first step is to insert a GOPTIONS statement into your program to choose the device driver: for example, `DEVICE=GIF`, `DEVICE=JAVA`, or `DEVICE=ACTIVEX`. In this paper, I will be using `DEVICE=JAVA`. Secondly, use the ODS HTML statement to specify the ODS options you will want to use. At a minimum, you must use the `BODY=` option to choose a body file to store the output created. The third step is to run a graphics procedure, such as `GCHART`, `GMAP`, `GLOT`, or `G3D` procedures. Finally, close the HTML destination by writing `ODS HTML CLOSE` in the program.

With utilizing the Java device and `PROC GCHART`, I am able to change the appearance of the bar charts in several ways. Users are allowed to do the following:

- Change the variables that are used in the chart and the roles they play in the chart
- Display the graph in 2-D or 3-D view
- Explore the graph by enabling the pop-up box that displays data values and displaying only a subset of the available bars
- Set drill-down options
- Change the shape or orientation of the bars
- Change all text colors, the background colors, and the outline color around bars
- Change the visibility or color of the surface behind a 2-D chart, or the surface behind, below, and beside a 3-D chart
- Hide or change the features of each axis and/or the legend
- Change the display of images in graphs that use images

ODS Statement Options

Before getting into options, I would first like to give you the syntax for the ODS HTML statement. So here it goes

ODS HTML *HTML-file-specification(s)*
<option(s)>;

The options for the ODS statement are endless. Below, I will list some of the basic options along with the ones that I used in my SAS program.

Important Information!!!

Documentation on these options can be found!!! While in SAS for Windows, from the menus, select Help> Sas System Help. Then select the SEARCH tab and search "applet parameters". From the list of found entries, select "GraphApplet Parameters."

The GTITLE and GFOOTNOTE options

The GTITLE and GFOOTNOTE affect the title and footnotes created. They control the destination of the title and footnote that are defined by the graphics program that generates the HTML output. They include all the currently defined titles and footnotes within the graphics output that is called by the body file. By choosing NOGTITLE and NOGFOOTNOTE, it tells SAS/GRAPH to suppress all the currently defined titles and footnotes from appearing in the resulting GIF file, but instead put the title and footnote in the associated HTML file. In other words, only when you open up the HTML file, will you see the title and/or footnote.

The GPATH option

The GPATH option tells SAS/GRAPH the destination for all graphics output that is generated while the HTML destination is open. You may want to save the GIF file in the same location as the HTML file you defined for the ODS output. If you do not use the GPATH option when creating a GIF or JPEG file, SAS/GRAPH will store the image in a sasroot folder. Without the GPATH option, because the HTML and the GIF or JPEG file are not created in the same location, there will be no link in the HTML file to the GIF or JPEG file. The result will be an icon that the browser uses in the place of the graphic image because browser cannot find it. It will not help to just move the graphics file to the same location as the HTML file, because there is no linkage to the HTML file. The GPATH option causes SAS/GRAPH to include the link from the HTML file to the graphics file.

The PARAMETERS= option

This particular option is important when using the Java, Javameta, and ActiveX device output. With this option you can control and customize things about your graph and its associated HTML file. The general form is

PARAMETERS=("parameter-name1"="parameter-value-1"... "parameter-name-n"="parameter-value-n")

This option writes the specified parameters between the <APPLET> and </APPLET> tags that generate dynamic graphics output. A thing to note is that the parameter name is not case sensitive, but the parameter-value is case sensitive for some of the parameters. There are numerous PARAMETERS= options to choose from which can be grouped according to their function. Borrowing from the SAS HELP documentation, I will list a few of the options by function available to the programmer.

A. Controlling the Background Appearance – affect either the background of the HTML file or the graphic image.

BackColor

This parameter specifies the background color of the HTML file. The default value is the default window color of the operating system.



BackDropColor

This parameter specifies the background color of all walls in the graph (including the back and floor). The default is white. The walls are the inside Y axes on a 3-D chart. The floor is the inside X axis on a 3-D chart. The back is the back of the chart on a 2-D or 3-D chart.

BackImage

This parameter specifies the URL of a background image to be used as the background for the graph or as an icon image. By default, no image is used and the background is drawn in a single solid color. If the URL begins with "http:" or "file:", the URL is assumed to be absolute; otherwise it is assumed to be relative to the HTML file you created with ODS.

DrawImage

This parameter specifies how the image specified using the BackImage parameter should be drawn. Valid values are "Center", "Position", "Scale", and "Tile". The default value is "Scale".

ShowGrid

This parameter specifies whether the axis grid on the backdrop for bar charts and scatter plots should be displayed. A value of "True" causes the grid to be displayed; a value of "False" causes the grid to be suppressed. The default value is "True".

B. Controlling Other Appearance Attributes – affect the appearance of the HTML file and graphics images in ways other than the background

Colors

This parameter specifies the colors that should be used for chart elements. The GOPTIONS COLORS= option has the same effect. You use this option by placing names of the colors you wish to use in the series as follows: "Colors"="Red Blue Green Yellow".

ColorScheme

This parameter specifies which predefined color scheme to use. By default no scheme is used. The GOPTIONS COLORS= option has the same effect, and will override the ColorScheme parameter. This parameter is a very handy way to select the colors for your charts. Available values are: Beach, Carnival, Colorprint, Commerce, Fall, Finance, Greyscale, Industrial, International, Meadow, Pastel, Primary, Southwest, Spring

OutlineColor

This parameter specifies the outline color for chart elements. For bar and pie charts this is the bar and pie slice outline color, that by default is the same color as the interior of the bar and pie slice. For scatter plots it is both the outline color of the markers as well and color of the line that joins plots when the plots are not subgrouped. The default outline color for scatter plots is gray. Instead of a color value, the special value "Same" can be specified. This value sets the color of the outline to the same color as the interior of the elements.

ShowLegend

This parameter specifies whether the legend should be displayed. A value of "True" causes the legend should be displayed; a value of "False" causes the legend to be suppressed. The default value is "True".

C. Controlling Drill-Down Functions – allows you to create HTML files with drill-down capability.

Ddleveln

This parameter specifies internal drilldown level specifications. Used with Local drill-down mode. There is no limit to the number of drill-down levels you specify.

Disabledrilldown

This parameter specifies that drilldown should be disabled for this applet. The value of the parameter does not matter, but a value must be specified as different browsers handle null values differently. Its presence indicates that drilldown should be disabled. In the GraphApplet for Java, the drill-down capability can again be enabled using the popup

menu. This allows the applet to dynamically generate all the drill-down information.

Drilldownfunction | Drillfunc

This parameter specifies the name of the JavaScript function to call when Script drill-down mode is in effect.

Drilldownmode

This parameter specifies the drill-down mode to use. Valid values are Any, HTML, Local, Script, and URL.

Any In Any mode, the applet first attempts to drill down locally (LOCAL). If that fails, it attempts to drilldown using JavaScript (SCRIPT). If both of these fail, the applet attempts to drilldown using URL mode (URL). If all of these fail, the applet attempts to drilldown using HTML substitution (HTML). As soon as any method succeeds, the sequence terminates.

HTML Uses a substitution pattern to dynamically generate a URL based on the selected chart elements. Attempts to load an HTML page whose URL is constructed by using the selected graph element's data values in a defined drill-down pattern. The pattern can be defined on the ODS HTML statement's DRILLPATTERN parameter.

Local Constructs a new chart based on existing data within the applet and replaces the old chart with a new chart containing the drilldown information. Attempts to drill down to a graph that organizes data according to the values represented by the selected graph element. This mode is only supported by the GraphApplet, which can dynamically generate the Local drill-down levels.

Script Invokes the user specified JavaScript function with information about the selected chart elements. The user function is expected to perform the requested drilldown action. The callback method determines how the passed information is used. To specify which method is called, use the ODS HTML statement's DRILLFUNC parameter. The callback method must be defined in the same HTML file that references the graph.

URL URL mode is provided for compatibility with previous SAS/GRAPH driver such as the GIF driver. It supports the old HTML= drilldown syntax.
The default is "Any".

Drillpattern

This parameter specifies the drill-down pattern to use when HTML drill-down mode is in effect. Described in more detail later in paper when Drill-down tags are discussed.

Drilltarget

This parameter specifies the drill-down window to use when HTML drill-down mode is in effect. The default is "_blank".

Helplocation

A string representing the URL from which the applet is to display help. The default is the help location for a particular applet release that SAS provides on its external Web site. The parameter can be either of the following:

- "Default" – this sets the location to the default location on the SAS external Web site.
- A valid URL – specified URLs are case sensitive. The URL should point to the location where your site has copied the applet help, which can be downloaded from the SAS external Web site. For details, see the SAS/GRAPH Web graphs documentation, which can be accessed from the Publications page on the SAS external Web site.

The value for this parameter is case sensitive.

Locale

A locale parameter specifies the language and country to use when displaying locale-sensitive strings. The default is to use the region settings of the browser's host machine. This parameter can be used to override the browser's default locale.

Patternstrip

This parameter removes the preceding and trailing white space from the drilldown substitution patterns before adding them to the generated drill-down string. The parameter supports the following modes; the default is FALSE:

"YES" "Y" "TRUE" "T" "STRIP" "S"	Remove preceding and trailing white space from drilldown substitution patterns before adding them to the generated drilldown string.
"COMPRESS" "C"	Remove all white space, including embedded spaces.
"FALSE"	Do not remove any white spaces. This is the default.

If you omit the PARAMETERS= option and you do not implement the URL mode by creating an HTML variable and using HTML= option on the procedure that generates the graph, the drill-down behavior depends on the applet. The MapApplet disables the drill-down feature. However, the GraphApplet implements the drill-down mode ANY.

Drill-down Tags that identify variable roles in a graph.

By using the Java Graph applet, it constructs drill-down information by identifying the role of the variables in the graph and uses those values based on the role. The way the applet uses the values of the variables is dependent on the graph type and the current drill-down mode. Special drill-down tags are used to determine the name of the variable's role and the graph variable that is performing the role. The following table is a list of the drill-down tags that are available.

<u>Tag Name</u>	<u>Substitution Definition</u>	<u>Supported By</u>
G_COLOR	The name of the current color variable	Scatter Plots
G_COLORV	The value of the current color variable for the selected graph element	Scatter Plots
G_DEP	The name of the current dependent variable	All Charts
G_DEPV	The value of the current dependent variable for the selected graph element	All Charts
G_DEPTH	The name of the current depth variable	Vertical Bar Charts and Scatter Plots
G_DEPTHV	The value of the current depth variable	Vertical Bar Charts and Scatter Plots
G_GROUP	The name of the current group variable	Bar Charts
G_GROUPV	The value of the current group variable for the selected graph element	Bar Charts
G_INDEP	The name of the current independent variable	All Charts
G_INDEPV	The value of the current independent variable for the selected graph element	All Charts
G_SUBGR	The name of the current subgroup variable	Bar Charts and Scatter Plots
G_SUBGRV	The value of the current subgroup variable for the selected graph element	Bar Charts and Scatter Plots
<i>name</i>	The value of an explicit variable name for the selected graph element. Although a variable name in a SAS program is not usually case-sensitive, the variable name is case sensitive when the name is used as a drill-down tag. You must use casing exactly the way it was used to create the variable in the data set. Use PROC CONTENTS to see how the variable name is stored in a SAS data set.	All Charts

Unformatted values are substituted for the drill-down tags, unless an optional “,f” parameter is specified for the tag, which instructs the substitution to use formatted values. Formatted values cannot be used with the Local drill-down method. If a formatted value does not exist then the unformatted value will take its place.

You may also specify an optional “,n” parameter. Usually the substitution string is URL-encoded for browsers that do not support embedded white space in URL strings. If the “,n” parameter is specified, then the resulting URL string is not encoded. No white space should be between the “,f” and “,n” parameters.

D. Controlling General Functions – customize other features

IgnoreGroup

This parameter specifies that the chart should initially ignore any group variable. Initially the chart does not

use the variable to the group role even if that variable was specified in ODS.

IgnoreSubgroup

This parameter specifies that the chart should initially ignore any subgroup variable. Initially the chart does not use the variable assigned to the subgroup role even if that variable was specified in ODS.

Label_#element

This parameter specifies labels for various elements of the chart. The “element” can be one of the following (substituting the element name in the parameter “Label_#element”):

Category Label for the X axis for the chart.
This parameter will be overwritten by the AXIS statement LABEL parameter. Thus, it serves the same function.

Response Label for the Y axis for the chart.
This parameter will be overwritten by the AXIS

statement LABEL parameter. Thus, it serves the same function.

Series Label for the Series element, if one is defined.

Group Label for the Group element, if one is defined.

Subgroup Label for the Subgroup element, if one is defined.

For example: "Label_#Response"="1987 Salary (Sum)"

The ATTRIBUTES= option

The ATTRIBUTES= option is used in conjunction with SAS/GRAPH procedures and DEVICE=JAVA in the GOPTIONS statement. It writes the specified attributes to an <APPLET> tag that generates dynamic graphics output. For information on the attributes that the tag supports, refer to one of the many reference books that are available on HTML. The general form is

ATTRIBUTES=("attribute-name1"="attribute-value-1"... "attribute-name-n"="attribute-value-n")

The CODEBASE= option

The CODEBASE= option creates a file path that is used by the GOPTIONS devices. The CODEBASE file path option has two definitions, depending on the GOPTIONS device used.

For the JAVA device:

The CODEBASE file path points to the directory that contains the java applets (.jar files). If a CODEBASE file path is not specified, the SAS server generates a default CODEBASE file path that is based on the install location for the applets. The install location for the applets is recorded in the SAS option APPLETLOC and can be modified by the user. It is important to note that if the default path created by the SAS install is used, the HTML output file must be viewed on the machine from which HTTP:// reference to the installed applets so that the output can be viewed from any machine on the web.

For the ActiveX device:

If CODEBASE file path (including a required filename) is specified for an ActiveX control, the browser will attempt to install the control from the location (including the filename) specified in the CODEBASE file path. This happens if the control is not already installed on the user's machine. By default, the SAS server does not generate a CODEBASE file path for the ActiveX control. A typical use for this functionality is for a user to put the install setup for the control on the web. All web users can then generate their HTML output with a CODEBASE that points to this location. If another

user (who does not have the control installed) tries to view the output, the user will be prompted to install the control on their machine.

The STYLE= option

The STYLE= option specifies the style definition to use in writing the HTML files. Note you must specify the option for each destination. Your site may have its own customized style definitions that are used for all HTML and Printer output to give the output a consistent appearance. Some of the standard style definitions to choose from are: default, beige, brick, brown, D3D, minimal, printer, statdoc.

ABOUT DRILL-DOWN GRAPHS

Drill-down graphs have areas that are designated to links of additional output, such as other graphs or reports. The output can be on the same web page or it could be located on a different web page. If the additional output is another graph, that graph may have its own hot zones that link to other targets or links. The main thing about these types of graphs is that you can establish a series of links that drill down to more detailed information. This type of graph is shown in my example of the Childhood Immunization Status for Medicaid recipients. The bar chart shows the percentage of children that received the particular vaccination for a state fiscal year. Since the bar chart is a drill-down graph, with each bar linking to the particular state fiscal years information. Each of the hot zones for the graph is based on the specifications you make on the graphics procedure that is used to create the graph. You provide the link information by storing the paths to target output in a special HTML variable. Several requirements are needed in a SAS program to generate a drill-down graph. You must create an HTML variable to hold the HTML locations. Normally, you create one HTML variable for each level of drill-down to store the links for the graphic areas. For links created from the legend, you can use the same variable to store links for the values in the legend. Secondly, assign values to the HTML variable that you have created. These values are the link locations that connect the graph to its output. The third requirement is running a SAS/GRAPH procedure to create the drill-down graph(s). In the graph procedure, it must use the HTML= option, the HTML_LEGEND= option, or both to link the variable with the graph(s). The last thing is running the procedures that create the output, which is directed to the locations identified by the HTML variable.

Since in my example I am focusing on the HTML drill-down mode, I will describe in detail what it is actually doing. When it is in effect, selecting the chart elements tries to open a Web page URL address that is associated and constructed by using the

graph's elements data values in a defined drill-down pattern. The drill-down pattern was described above. In a SAS/GRAPH program you specify the HTML drill-down mode and drill-down pattern along with any other options in the ODS statement. When selecting the drill-down mode and pattern, you are constructing the URL addresses.

ODS html

```
parameters=("DRILLDOWNMODE"="HTML"  
            "DRILLTARGET"="_SELF"  
            "DRILLDOWNPATTERN"="{&g_indepv}.html")
```

By default with the HTML drill-down mode, a new browser window will open and the URL is loaded in the new window. To control whether the URL is loaded in the new window or in the current window, use the DRILLTARGET parameter. In the above example, the drill-down window will be the current one. Other valid targets are "_blank", "_parent", and "_top".

The URL in the above ODS statement will be built using the value of the current independent variable for the selected graph element. The independent variable must be among the variables used to generate the JAVA bar chart.

EXAMPLE

See Appendix for a taste of the SAS code used to produce the drill-down charts and reports. The whole program will be available as a handout at the conference, along with a demonstration of how it all works.

CONCLUSION

Reports and bar charts are basic tools used to show and display information for and results of a given study. Adding a few lines of SAS code, you can take these basic tools and turn them into advanced and fascinating results. Using the ODS HTML statement and its options, allows you to create your own or your company's web enabled graphs and reports.

REFERENCES

SAS Online Documentation Version 8.2

ACKNOWLEDGEMENTS

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

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

CONTACT INFORMATION

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

Jessica Bonow

Arkansas Foundation for Medical Care

401 West Capital, Suite 508

Little Rock, AR 72201

Work Phone: 501-375-5700

Fax: 501-375-5705

Email: jbonow@afmc.org

APPENDIX

Examples of SAS Code to produce charts, create reports, and incorporate drill-down functionality for the results of Childhood Immunization Status.

```
*****;
** Begins the ODS output for SFY 2001 data;
** Uses the ODS options that were discussed earlier in the paper;
ods listing close;
ods html file="sfy2001.html" gpath=odsout &codebse
  parameters=("disabledrilldown"="true");

** creating sfy01 data set;
data sfy01;
  set results_3;
  if project='SFY2001';
run;

title1 "Childhood Immunization Status - SFY 2001";
title2 "ARKids A";
pattern1 c=ligr; pattern2 c=ligr;
pattern3 c=ligr; pattern4 c=ligr;
pattern5 c=ligr; pattern6 c=ligr;
pattern7 c=ligr;

proc gchart data=sfy01;
  format vaccine $vac.;
  vbar3d vaccination / sumvar=measure patternid=midpoint discrete
  footnote f=zapf h=.7 j=1 'Combination 1 - 4 DTaP, 3 Polio, 1 MMR, 2 H
    Influenza, 3 Hepatitis B ';
run;
quit;

** Ends ODS output;
ods html close;

*****;
*****;
** Begins the ODS output for SFY 1998, SFY 1999, SFY 2000, SFY 2001;
** and National data;
** Uses the ODS options that were discussed earlier in the paper;
ods html path=odsout file="java_cc_immun_sfy98_99_00_01.html" &codebse
  parameters=("drilldownmode"="html"
    "drilltarget"="_self"
    "drillpattern"="{&g_indepv}.html")
  gtitle style=statdoc;
  legend1 label=none position=(bottom center) shape=bar(3,2) value=none;
  axis1 label=none order=(0 to 100 by 25) value=(h=5pct f=zapf);
  axis2 label=none value=none;
  axis3 label=none value=(h=4pct f=zapf);
  pattern1 v=s c=cx5A58A6; pattern2 v=s c=cxFFCC99; pattern3 v=s c=grpb;
  pattern4 v=s c=bwh; pattern5 v=s c=ligr;
  title j=c h=6pct c=black f=zapf 'Arkansas Foundation for Medical Care';
  title2 j=c h=4pct c=black f=zapf 'Childhood Immunization Status';
  title3 j=c h=4pct c=black f=zapf 'ARKids A';
  footnote f=zapf h=1 j=1 'Combination 1 - 4 DTaP, 3 Polio, 1 MMR, 2 H
    Influenza, 3 Hepatitis B';
```

```

proc gchart data=results_3;
format vaccine $vac.;
vbar3d project /discrete group=vaccine subgroup=project sumvar=measure
        legend=legend1 gaxis=axis3 maxis=axis2 raxis=axis1
        patternid=subGROUP autoref;

run;
quit;

*****;
*****;
** Creates a report from the results_3 data set that gives the information
for SFY 2001 ;
data report_6;
    set results_3;
    if project='SFY2001';
run;
title;

proc report data=report_6 nowd headline
        style (header)=[FONT_FACE='Arial Narrow'
                        FONT_WEIGHT=BOLD
                        BACKGROUND=cx5A58A6];
column vaccination num den measure;
define vaccination / display width=55 "IMMUNIZATION"
style=[BACKGROUND=cxFFCC99];
define num / analysis format=10. "NUMERATOR" ;
define den /display format=11. "DENOMINATOR";
define measure /display format=9.2 "MEASURE";
compute before _page_ / LEFT
        STYLE=[PREIMAGE='s:\webpage maps\AFMClogo3.GIF'
        FONT_WEIGHT=BOLD
        FONT_STYLE=ITALIC
        FONT_SIZE=3
        FONT_FACE='TIMES'];
line " ";
line @20 "ARKids A Immunization Results";
line @20 "For State Fiscal Year 2001";
endcomp;
format vaccine $vac.;

run;

```