

Paper 089-2007

Creating ActiveX Graphs for Presentations using SAS® Enterprise Guide® ... And Drill-Down Graphs Too

Terek Peterson, Shire, Wayne, PA
Robert Gordon, Cephalon, Inc., Frazer, PA

ABSTRACT

This paper introduces the SAS® user to the basics of using SAS Enterprise Guide® (EG) to create graphs. The example project will show how to pull data from ClinTrace® Oracle tables to make datasets, how to subset and merge the datasets with the point and click EG interface, how to produce a 3D ActiveX HTML Graph, and how to insert the graph into a PowerPoint presentation. Using the ActiveX graph toolbar, it will be demonstrated how to pivot, size, and zoom the graph within PowerPoint or Excel. A discussion and illustration will explain some of the limitations of the EG graphing capabilities. Exporting the code generated from the example project in EG allows enhancements with SAS using the TEMPLATE procedure and SAS/GRAFH code to produce publication quality output. Lastly, the authors will show how to take advantage of the ODS HTML output destination to create a drill-down series of graphs of increasing detail, which end in an Excel spreadsheet of the data.

INTRODUCTION

In the pharmaceutical industry, visual and graphic displays are important analysis and presentation tools. The development of user-friendly interactive graphics would greatly enhance the usefulness of these tools. By utilizing the power of SAS, production of high-powered, presentation quality graphics is possible. More recently, data-exploration has become a desired specification. To address this, the enhanced graphs via ActiveX Control allow for this interactivity. The data analyzed for this project is in a ClinTrace® Oracle database used for data capture and reporting. Therefore, it was not inherently useful for graphic output. By using SAS/ACCESS to Oracle, SAS Enterprise Guide (EG) could easily access the data. A subset of the data was then used to create interactive high-powered graphics that allows the user to cut and paste the graphic into various Microsoft based programs. As an additional experiment, the authors show how to create a drill-down series of graphs for better data connectivity. To afford understanding, background on the data and tools is provided. The paper will then discuss, in part, the drill-down capabilities of SAS.

ABBREVIATIONS

EG	SAS Enterprise Guide®	ODS	Output Delivery System
HTML	Hypertext Markup Language	PT	Preferred Term
MedDRA®	Medical Dictionary for Regulatory Activities	SOC	System Organ Class

BACKGROUND AND DEFINITIONS

The data analyzed is post-marketing safety data owned by a pharmaceutical company. This data represents safety information about products on the market. Although the data has numerous attributes and characteristics, an understanding of the structure behind the safety data is most important to understand the graphics, techniques, and ultimately the drill-down characteristics. The data is adverse event terms reported in conjunction with the use of a pharmaceutical product (i.e. headache, nausea etc.). These terms are part of a hierarchical architecture known as the medical dictionary MedDRA®. The dictionary itself has five levels, with each level having more granularity than the previous level and subsumed under the previous. The analysis and graphics focus on 2 of these levels. The highest level of MedDRA®, the System Organ Class (SOC), has 26 categories relating to major body systems (i.e. Nervous SOC, Cardiac SOC). The next level of interest is actually the fourth tier and is the Preferred Term (PT). The PT level has approximately 16000 different entries.

The data resides in a post-marketing safety database. The Clintrace® application allows for post-marketing data to be captured and reported. This Oracle based application is a comprehensive database software package for reporting and tracking clinical adverse events. Output from this application is limited to report format only and virtually no graphics.

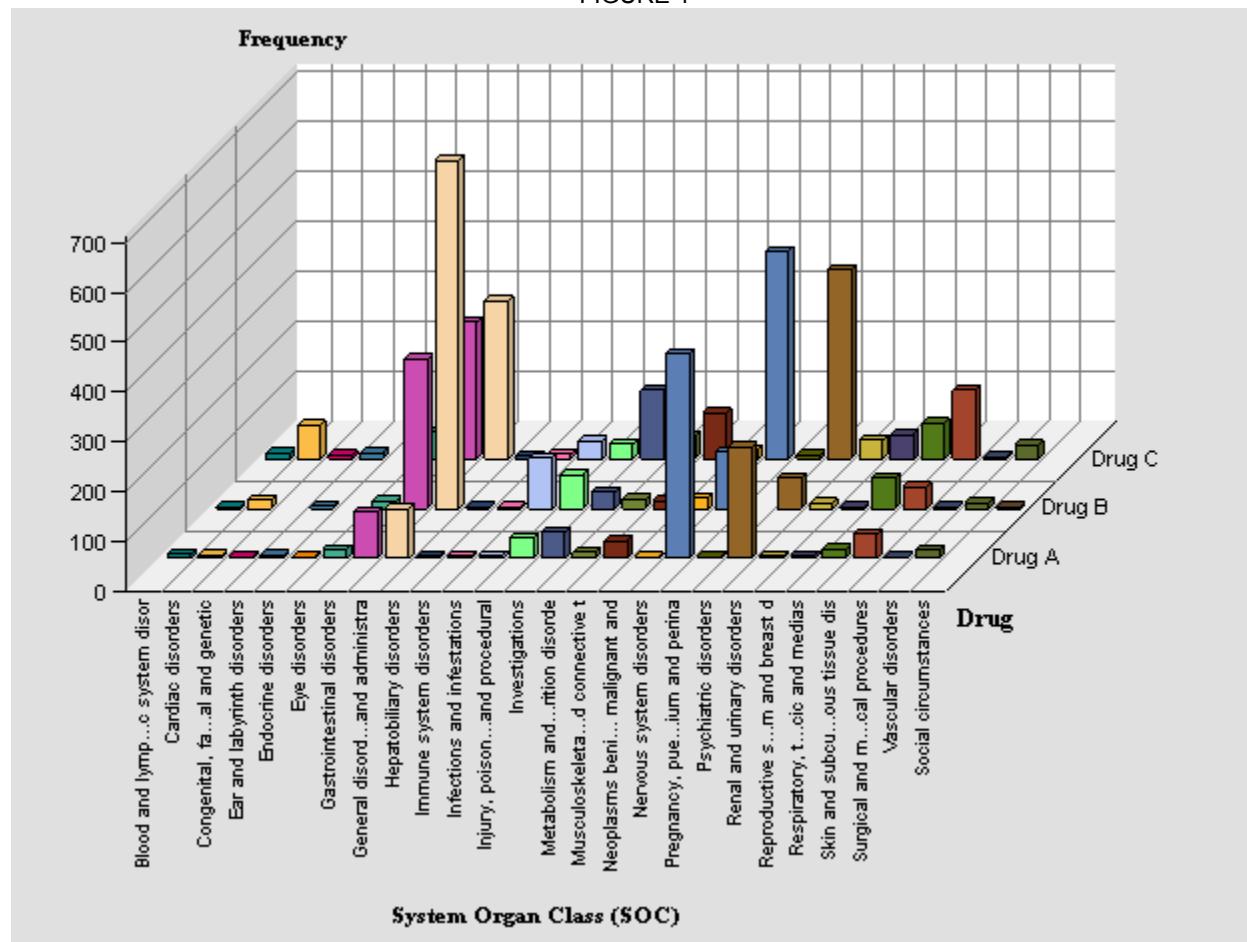
Enterprise Guide® (EG) is the SAS tool used for the bulk of the project. EG, in very simple terms, is SAS's visual interface tool that allows users to easily perform point and click analysis with SAS. It would seem, most of the functionality performed in Base SAS, SAS/STAT, and SAS/GRAFH can be performed in EG with point and click and wizard tools. The beauty of EG is that the user needs to enter little to no traditional SAS code. EG writes all the code for a project that, in turn, can be altered or exported for further enhancement. EG also has various output formats including RTF, PDF, HTML, XML, and ActiveX. For this project, interest lies in the ActiveX output option. ActiveX

control is an interactive graphic format that embeds data into the graphic. Using the ActiveX plug-in, the end user can interact with the graphic. Numerous visual adjustments as well as access to the data are possible.

For the purpose of this project, a working definition of the term drill-down would denote the ability to create interactive graphics that give the user the ability to click on the graphic to explore numerous other levels of the data. A click on a bar, or line, or data point in one graph will open an additional interactive graph with details based upon the data point chosen. For example, one may start with a graph of the United States that has various statistics. By clicking on a state, an additional graph opens with statistics concerning that individual state. A click on a county in the state may open an additional graph and so on. The levels are only constrained by the amount of necessary programming.

Entry to the following project began with the SOC levels of all cases. Through data points on the initial object, drill-down capabilities led to individual cases and to the development of case series in the form of Excel spreadsheets.

FIGURE 1



THE PROJECT

Using SAS EG, the steps needed to create an interactive 3D graphic are straightforward. They involve importing the data, querying the data and creating the graph. This project focuses on the creation of a 3D interactive graph involving display of the PT frequency for three different products broken down by SOC (figure 1).

The first step in any project is accessing the data. Since the data resided in Clintrace®, a library reference was used in EG to access the data through SAS/ACCESS to Oracle. As SAS users already know, SAS permits the user to access data from numerous data formats, so EG will too. This data step allowed real-time, read-only access to the entire Clintrace® database. Once the data was imported, tables (datasets) from Clintrace® were chosen that were going to be utilized in the project. The following one line statement used to connect to the database represents the simplicity of using SAS/ACCESS to Oracle get to the Oracle data:

```
libname xxxxx oracle user="xxxxxx" pw="xxxxxx" path="xxxxxx" schema "xxxxxx";
```

The next step is to subset the data into the final dataset that will be used to produce the graph. The final dataset is produced by using SAS EG's querying tool to select, sort, filter and query the data. First, however, the tables or datasets that were imported must be joined. This is accomplished in EG with a process that resembles the process used in Microsoft Access® (see figure 2). EG will join tables by default but it is imperative that the user verifies the joins. This step is extremely important and stresses the need to have intimate knowledge of the data.

FIGURE 2

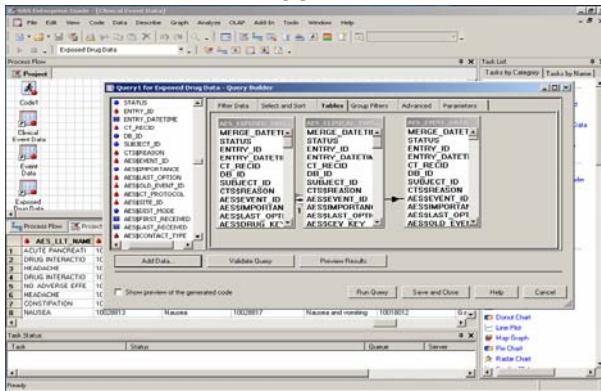
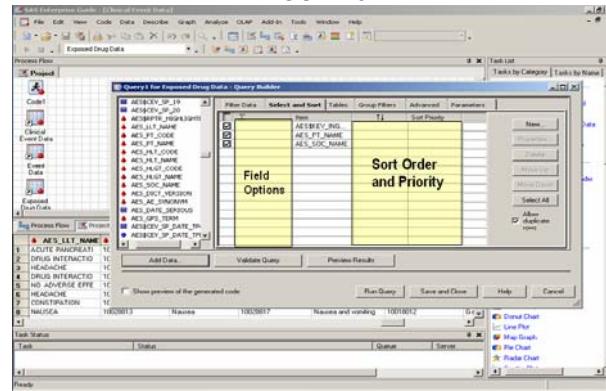


FIGURE 3



Selecting the variables (columns), sorting the selections, and filtering the data are accomplished with the aid of the query builder in EG (see figure 3). The query builder offers a myriad of options including sort order, priority, column analytics, and an almost exhaustive list of operators to choose from in the filtering window.

A nice attribute of EG is the ability to display all of the existing column values to choose from in the filtering process. Specific values are chosen using various operators during this step (see figure 4). At this step, EG has subset the initial data into the final dataset and the graphing process can begin. It is important to note that the project is still at a surprisingly basic ‘point and click’ mode with EG writing code behind the project as well as developing the project flow chart in the project window as shown in figure 5.

FIGURE 4

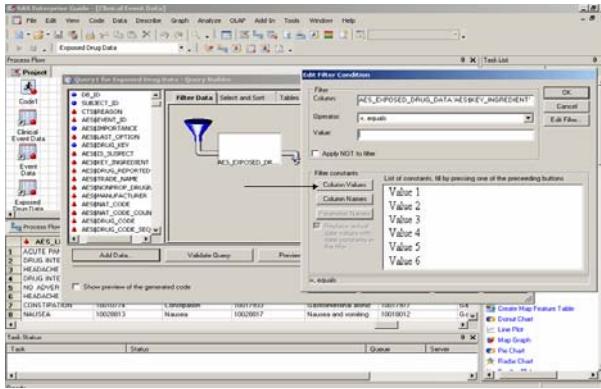
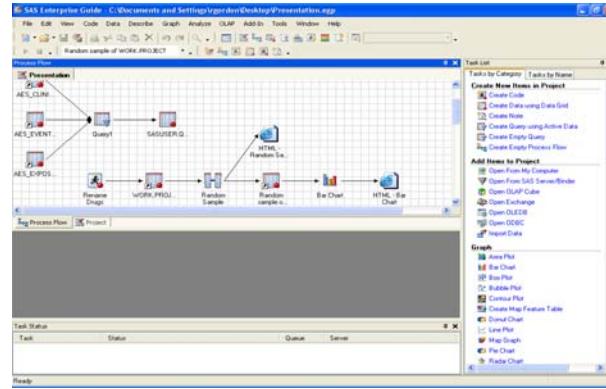


FIGURE 5



Throughout this entire project, no other code has been written except for the library reference statement. This is not to say that in-depth customization is not a capability. Code generated by EG is standard SAS code and can be altered or enhanced to develop your project. As shown in figures 6, 7, and 8 on the next page, the final graphic was developed using a wizard like most processes in EG. In figure 6, a 3D bar chart was chosen from the large variety of graphics available. In figure 7, the variables were chosen for the vertical bars and horizontal rows. In figure 8, it is shown that enhancements can be added to the graph like formatting the axis, choosing a font, and adding a title. Many other enhancements are available, but not all that might be available via the procedure in SAS/Graph.

FIGURE 6

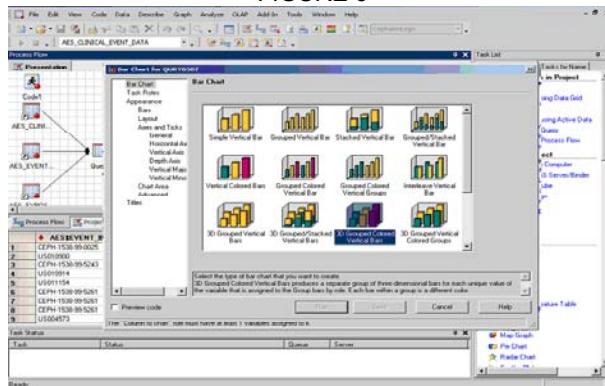


FIGURE 7

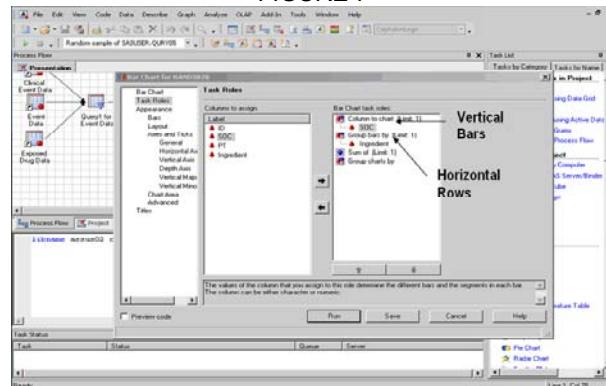


FIGURE 8

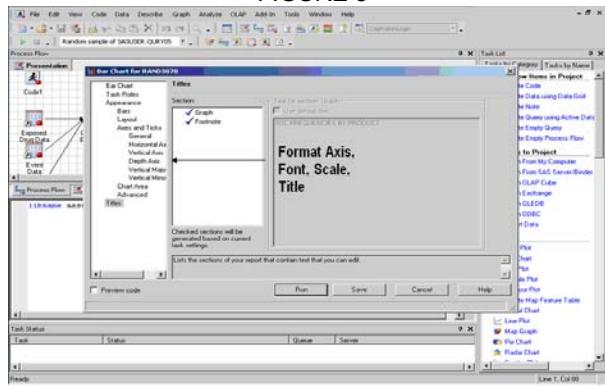
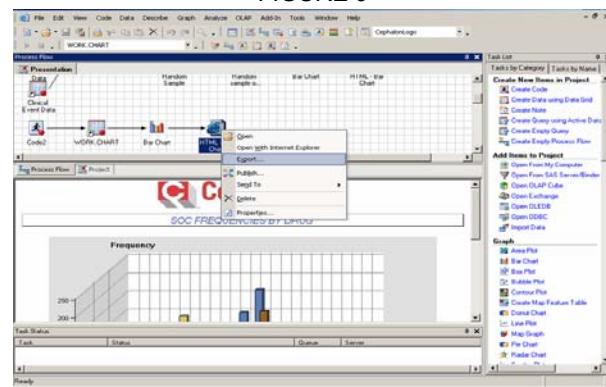


FIGURE 9



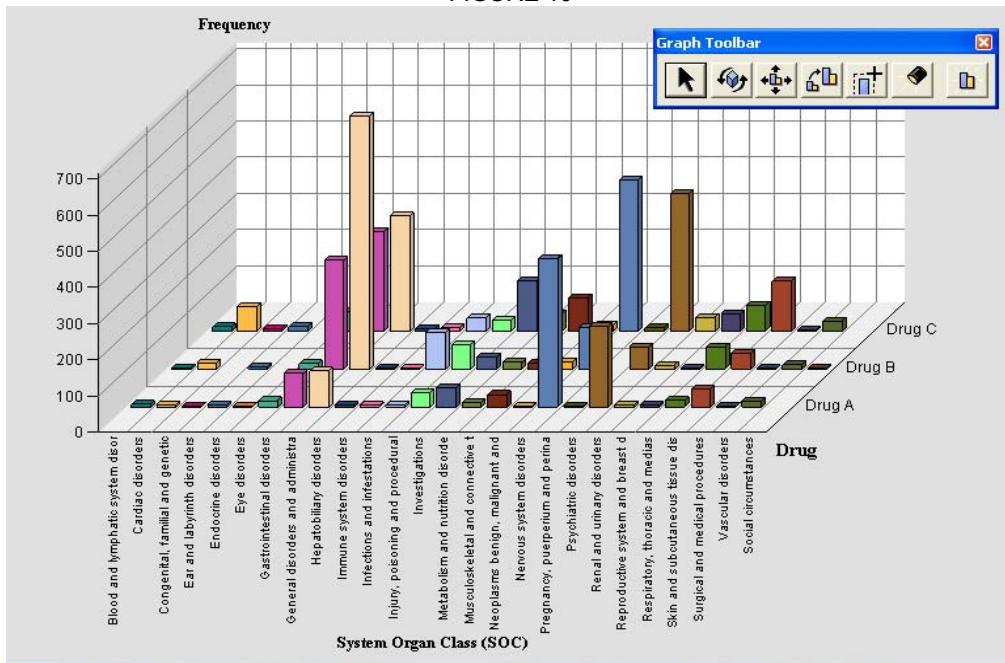
So far the project has included accessing the data through SAS/Access to Oracle, importing the data, querying, filtering, subsetting and graphing the data through EG. The final graphic is ready to be exported into a more suitable program for presentation or analysis purposes. A simple right click on the graphic will allow the user to export the graphic to a desired location in the output format. One can also cut and paste the interactive graphic at this point (see figure 9 above). Once exported or copied, the user has the ability to interact with the graphic using the ActiveX tools (see figure 10 on the next page). The free SAS/GRAFH ActiveX plug-in from SAS is the only application required to be able to view and use the ActiveX functions within Microsoft products. Additionally, the data for the graphic is embedded in the file and can be viewed through the ActiveX toolbar.

During this project, a presentation quality, high-powered graphic that can be manipulated through the ActiveX application has been created. Interest lied in the possibility of creating drill-down graphics through EG. In short, the SAS system has the ability to create very functional and impressive drill-down systems. A drill-down system that links through the highest level of MedDRA® to the actual case-series in spreadsheet form was created working in both EG and SAS. A SAS macro, PROC TEMPLATE, and ODS HTML code were necessary steps for this process.

GRAPHIC TOOL AND FUNCTIONS AVAILABLE

It is worth exploring the two topics noted above: the use of a graphic toolbar and the ability to interact with the data. By right clicking and choosing 'Graph Toolbar', the interactive toolbar appears as shown in figure 10 on the next page.

FIGURE 10



Seven individual tools appear on the toolbar. From left to right these tools are the following:

- a typical mouse pointer
- a rotational tool with the ability to rotate the graph in all directions
- a tool allowing the user to move the graph within the window
- a zoom tool manipulating the entire graph
- a precise zooming tool allowing the user to choose a specific area to enlarge
- a tool that allows the user to highlight specific areas
- a tool which enables the user to return the graphic to its original form.

Another right click will give the user the choice to change the chart type from a bar chart to a pie chart as well as open the properties pane as shown in figure 11.

FIGURE 11

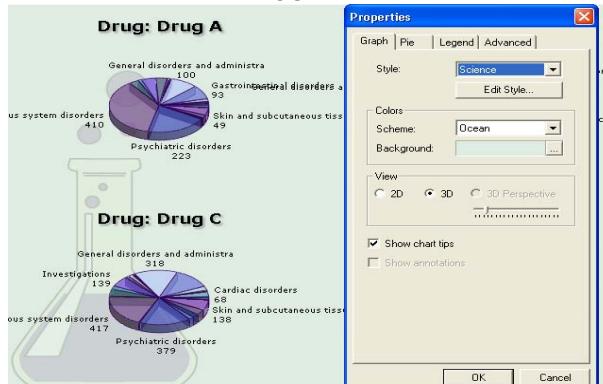
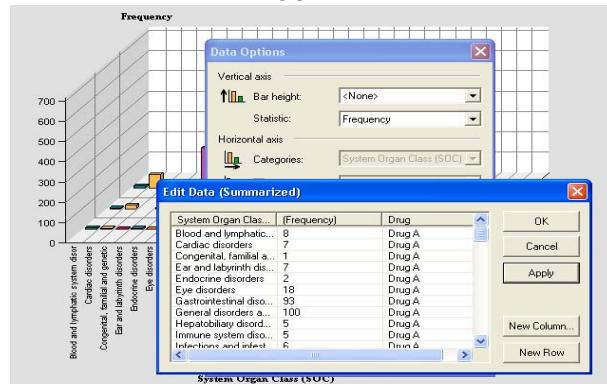


FIGURE 12



The four tabs of the properties pane offer the user the ability to customize the graph's appearance, change the colors of the pie slices or bars, alter axis appearance and labels, and select some additional lighting, sizing, and appearance options. An important additional feature of the properties pane is the ability to access the data behind the graphic. As seen in figure 12 on the previous page, one has access to the underlying data that used to populate the graph, an interesting feature of the ActiveX environment. The ability to see the underlying data as well as edit the data to create updated or different graphs gives the user more flexibility. Various scenarios and 'what if' examples can be explored directly through the ActiveX environment.

CONFIGURING THE DRILL-DOWN MODE

Now that EG creates the graphing code, how can hyperlinks between graphs and spreadsheets be created? EG does have capabilities to do this, however the number of hyperlinks and documents created required the use of the SAS Macro language. SAS OnlineDoc explains that the URL drill-down mode uses the `HTML=` option to name a link variable that provides drill-down URLs. This mode is implemented in a similar manner to the type of drill-down that is available for the GIF, JPEG, and PNG device drivers, except that in this case, the ActiveX control associates drill-down URLs with graph elements without using an image map. The six basic steps to create a set of drill-down graphs that are hyperlinked to each other follow.

STEP 1 – SETUP GOPTIONS

Specify the device driver by choosing ActiveX as the device. Set any other global options. In this example, the `GOPTIONS` statement resets the `GOPTIONS`, sets the device driver to ActiveX and specifies the size of the graphs created.

```
goptions reset=global device=activex ypixels=600 xpixels=800;
```

STEP 2 – CLOSE ODS LISTING DESTINATION

As per SAS OnlineDoc, close the ODS Listing destination to conserve resources.

```
ods listing close;
```

STEP 3 – START ODS HTML WITH URL DRILL-DOWN MODE

Open an HTML output file in ODS, choose a style, and specify URL drill-down mode. See the section Custom Style Using PROC TEMPLATE in this paper for PROC TEMPLATE code examples.

```
ods html body=drugs path=urldrill style=Custom parameters=( "drilldown"="url" );
```

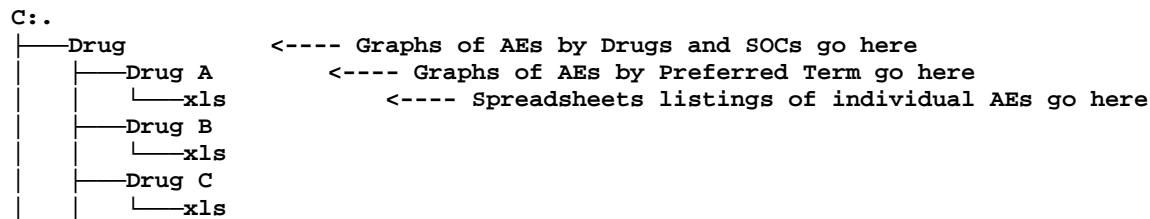
Note the applet parameter `DRILLDOWN=URL` does not need to be specified. This drill-down mode is invoked by default when the `HTML=` options is used in the SAS/GPGRAPH procedure used to generate the graph.

STEP 4 – CREATE THE HYPERLINKS

Specify drill-down URLs by adding a link variable to the dataset. Add links to the graph with the `HTML=` option. Create variables in the dataset containing the Adverse Event data variables with the directory and name of the HTML file to link. It is important that this link will become the actual name and location of the HTML document produced. In the following example, the variable `DRUGLINK` containing the hyperlink is created which will link the bar in the overall graph to the chart showing all the different SOCs for that particular drug.

```
data aelinks;
  set allae;
  druglink='href="C:\Drug\'||trim(drug)||' AES by SOC.html"';
  soclink='href="C:\Drug\'||trim(drug)||'\''||propcase(trim(soc))||'
    ' AES by Preferred Term.html"';
  ptlink='href="C:\Drug\'||trim(drug)||'\xls\'||propcase(trim(pt))||'.xls"';
run;
```

The graphs and spreadsheets are placed in the appropriate directories as shown below. The overall drug AEs by SOC graphs go into the C:\DRUG subdirectory. The particular drug's AEs by preferred term go into the subdirectory for that drug, for example C:\DRUG\DRUG A subdirectory. Finally, individual spreadsheets listing the AE data for each drug's preferred term go into XLS subdirectory, for example C:\DRUG\DRUG A\xls subdirectory



STEP 5 – SPECIFY SAS/GGRAPH PROCEDURE WITH HTML= OPTION AND CREATE SPREADSHEETS

Specify a SAS/GGRAPH procedure to generate the graphs. Specify the statement option HTML= to identify the link variable. The example code that follows uses the GCHART procedure. Notice the procedure uses the variable DRUGLINK created earlier with the HTML= option to construct the hyperlink.

```

title1 "Total Adverse Events by Drug";
footnotel "Generated on %SYSFUNC(DATE(), EURDFDE9.) at %SYSFUNC(TIME(), TIMEAMPM8.)";

proc gchart data=allae;
  hbar3d drug / frame descending freq cfreq percent autoref clipref
  patternid=midpoint html=druglink;
run;

```

Specify a procedure like PROC REPORT or PROC PRINT to generate a spreadsheet. Since ODS HTML is the ODS destination, it is easy to create the spreadsheets too. SAS 9.0 and above provide direct, transparent access to Microsoft Excel. The following code produces a spreadsheet through PROC REPORT and ODS HTML.

```

ods html body="Drug\Drug A\xls" path=urldrill style=Gears
  parameters=("drilldown"="url");

title1 j=left color=black height=16pt "&drugname Adverse Events of";
title2 j=left color=black height=16pt "&ptname";

proc report data=drill nowd headline headskip split="\"
  style(report)=[cellspacing=2 borderwidth=2 bordercolor=cx737373]
  style(header)=[foreground=cx00407A font_size=12pt borderwidth=4]
  style(column)=[foreground=cx00407A font_face=Arial font_size=10pt]
  style(lines)=[foreground=black background=black font_size=10pt];

  column event frd source gender age serious narrate;
  define event / display "Case Number" width=30;
  define frd / display "Case First\Received Date";
  define source / display "Source\of Case";
  define gender / display "Gender";
  define age / display "Age";
  define serious / display "Serious";
  define narrate / display "Narrative" width=100 flow
style(column)=[cellwidth=6in];
run;

```

STEP 6 – CLOSE ODS

Close the HTML output file and reopen the ODS listing destination.

```

ods html close;
ods listing;

```

CUSTOM STYLE USING PROC TEMPLATE

The default style used by EG produces good-looking graphs. To produce publication or presentation quality graphs, a custom style should be created. This paper will not show how to use PROC TEMPLATE, because there are many excellent papers on customizing and using templates. The code that follows will simply show some of the customizations used to make this graph standout.

START WITH AN EXISTING STYLE

The TEMPLATE procedure is a difficult one to understand, so start with an existing template and modify. This code creates a new template called Custom based off the provided SAS template Gears.

```
proc template;
  define style Custom;
    parent = styles.Gears;
```

MAKE FONT CHANGES

Change the template to use the font best suited for the graph. Some companies have corporate standards regarding font usage, put that font here. The size of the font and font face can be specified.

```
replace fonts /
  'TitleFont2' = ("News Gothic BT Roman, Arial, sans-serif",12pt,Bold)
  'TitleFont' = ("News Gothic BT Bold, Arial, sans-serif",14pt,Bold)
  'StrongFont' = ("News Gothic BT Roman, Arial, sans-serif",12pt,Bold)
  'EmphasisFont' = ("News Gothic BT Light, Arial, sans-serif",8pt,Italic)
  Etc...
replace GraphFonts /
  'GraphDataFont' = ("News Gothic BT Condensed, Arial, sans-serif",6pt)
  'GraphFootnoteFont' = ("News Gothic BT Condensed, Arial, sans-serif",6pt)
  Etc...
```

MAKE COLOR CHANGES

Change the template to use the colors again best suited for the graph. Companies have corporate standards regarding color usage; the corporate communications department can provide a list. There are multiple ways to define what color to use via color-naming schemes. Some valid color-naming schemes include RGB, CMYK, HLS, HSV and Gray-Scale color codes; the example code below uses RGB color codes. The RGB color-naming scheme specifies color in terms of its red, green, and blue components. Color names are of the form CXrrggb.

```
replace colors /
  'headerfgemph' = cxDEE385
  'headerbgemph' = cxCF142B
  'headerfgstrong' = cxDEE385
  'headerbgstrong' = cxCF142B
  'headerfg' = cxDEE385
  'headerbg' = cxCF142B
  Etc...
replace GraphColors /
  'gcerror' = cx000000
  'gerror' = cxB0C4AE
  'gcpredictlim' = cx2E522D
  'gpredictlim' = cxB0C4AE
  'gcpredict' = cx2E522D
  'gpredict' = cx2E522D
  'gcdata3' = CXA8B508
  'gcdata2' = CXDEE385
  'gcdata1' = CXDEE385
  Etc...
```

FINAL TOUCHES – TRANSPARENCY AND BACKGROUND IMAGE

When making a 3-D graph with multiple items as the Z-axis, bars that are in front can hide the bars displayed in the back. By making the bars somewhat transparent, these other bars are then visible. In the code below, adjustments to the GraphCharts attribute changes the transparency of the bars. Adding a background image is simple to do, just point to the image within the GraphBackground attribute like shown below. Now, there is a great image behind the

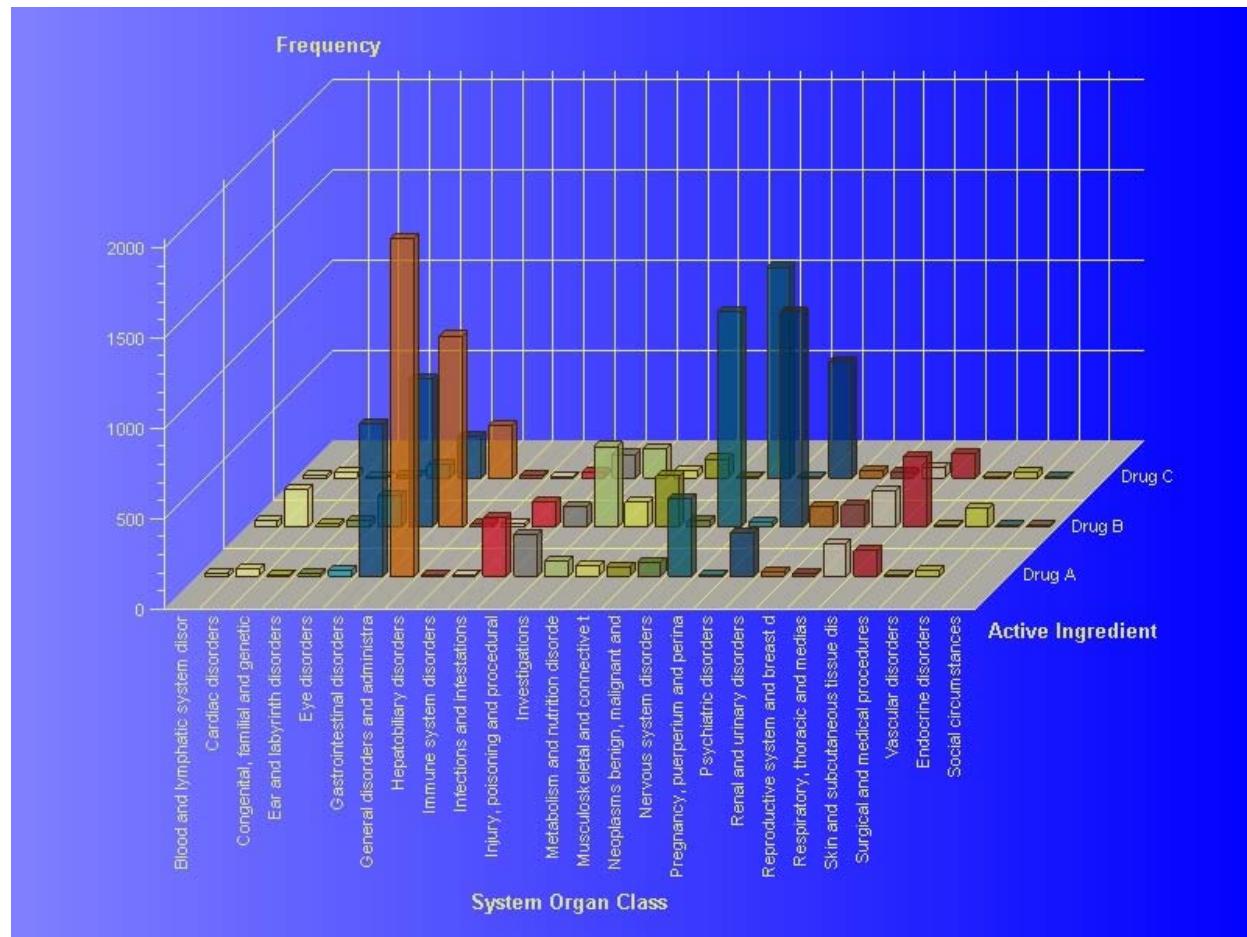
graph but the graph's grids cover the image up too much. To remedy this situation, use the GraphWalls attribute and set transparency to one. The 1.0 has the meaning of 100% transparent.

```

style GraphCharts from GraphCharts
  "Chart Attributes" /
  transparency = 0.25;
replace GraphBackground
  "Graph background attributes" /
  background = colors('docbg')
  image = "X:\EG\BACKGROUND\PICTURE.JPG"
  vjust = T;
style GraphWalls from GraphWalls
  "Wall Attributes" /
  transparency = 1.0;
end;
run;
```

THE FINAL RESULT

The final result below is informative, interesting, flexible, and portable. The graph is more than just a pretty picture; it is a presentation and information conveyance tool.



CONCLUSIONS

LIMITATIONS

Knowledge of SAS coding is useful but not a requirement for EG however, intermediate technical skills seem a necessity to produce high quality, meaningful output efficiently and effectively. Knowledge of SAS is an obvious advantage and gives the user the ability to edit EG generated code. In fact, publication quality graphics may require editing of the generate code. The advanced features and code generating capabilities make EG much slower to run programs than traditional SAS, however, EG is writing the code.

CAPABILITIES

EG is an intuitive, user-friendly windows interface that can readily access numerous data sources. The ability of EG to write the code for the project is an obvious advantage and a time saving attribute. The ability to manipulate the code is an additional advantage.

The overall environment in EG is very intuitive and user-friendly. The viewing windows are easily resized, hidden and manipulated to suit the user's needs. The flow chart project window is very useful for versioning, audit trail work and 'snapshot' in time applications as well as a visual representation of the project flow.

EG is a formidable tool for presentation quality graphics and prototyping output. An experienced SAS user teamed with EG will be able to create high-powered analysis and publication quality graphics. ActiveX control used in the EG environment allows for powerful manipulation and visualization of graphic output.

REFERENCES

Bruns, Dan (2005), "An Introduction to the Simplicity and Power of SAS/Graph", *Tutorials of the Thirtieth Annual SAS Users Group International conference*, Cary, NC: SAS Institute

"Introduction and profile MSSO. MedDRA®: How it all started" 2003

<http://www.MedDRAmsso.com/NewWeb2003/docs/7165-120%20MSSO%20Introduction%20datasheet.pdf> (2005)

Ravenna, Andy & Syphus, Stacey (2005). "Querying and Reporting Using SAS Enterprise Guide Course Notes", Cary, NC: SAS Institute

SAS Institute Inc. (2003), "SAS/GPGRAPH Colors and Images: Specifying Colors in SAS/GPGRAPH Programs", Cary, NC: SAS Institute

SAS Institute Inc. (2003), "Summary of ActiveX and Java Support: PROC GCHART", Cary, NC: SAS Institute

SAS Institute Inc. (2003), "Creating Interactive Output for Java: Configuring Drill-Down Links for Java and ActiveX", Cary, NC: SAS Institute

SAS Institute Inc. (2003), "Creating Interactive Output for Java: Examples of Interactive Java Output", Cary, NC: SAS Institute

SAS Institute Inc. (2003), "SAS/GPGRAPH Statements: Example 10. Creating a Bar Chart with Drill-down for the Web", Cary, NC: SAS Institute

SAS Institute Inc. (2003), "SAS/GPGRAPH Statements: AXIS Statement", Cary, NC: SAS Institute

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

Terek J. Peterson, MBA
Shire
725 Chesterbrook Blvd.
Wayne, PA 19087
484.595.8947
tpeterson@shire.com

Robert Gordon, MS
Cephalon, Inc.
41 Moores Rd
Frazer, PA 19355
610.738.6720
rgordon@cepalon.com