**What's In A Map?**

**A Macro-Driven Drill-Down Geo-Graphical Representation System**

Louise Hadden, Abt Associates Inc., Cambridge, MA

---

**ABSTRACT**

This paper presents a macro-driven system to create drill-down maps and present them on intranets or the internet, using BASE SAS® and SAS/GRAPH®. This system can be used with SAS V8 (or SAS V612 in conjunction with the %ds2htm macro supplied by the SAS Institute for HTML only) in any operating environment, including Red Hat Linux, Windows and AIX/UNIX. Although the primary focus of the paper is on interactive HTML map output, some different graphic outputs (for PC SAS only) will be briefly demonstrated. Presentation graphics contained in HTML, XML and PDF files are static and require only an internet browser and/or Adobe Acrobat to view. They are quickly and easily updatable on the host system even if the server does not have SAS installed. Other presentation options SAS provides such as JAVA and Active-X are interactive, but require specific versions of SAS to be installed on the host system to allow interactivity.

---

**INTRODUCTION**

Since I first started using a beta version of SAS on an IBM 360 at MIT to analyze data for a thesis more than 20 years ago, SAS has evolved into an amazing computing tool. The purpose of this paper is to outline a simple method to take advantage of the new and improved graphic functionality in SAS V6 using some old tried and true tools such as SAS Macro Language and user-defined formats, and some new methods provided by the Output Delivery System (ODS). Some of the presentation options demonstrated in the paper are specific to PC-SAS; others can be used and have been tested on other platforms including AIX UNIX and Red Hat Linux.

**KNOW YOUR DATA (GETTING UP CLOSE AND PERSONAL WITH SAS-SUPPLIED MAP DATA SETS)**

The map-related data sets provided by SAS have always seemed somewhat mysterious to me, possibly because as an "old dog" I have not been accustomed to thinking of computing in a graphic way. From paper tape and punch cards to GUI is a long way to travel! My approach to understanding the map data sets is the same approach I use with any "unknown" data set; do a proc contents and test print some observations and take a look.

SAS provides maps and map-related data bases for geographical areas all over the world within the SAS/GRAPH package. In the US these maps are free of charge down to the STATE COUNTY level. Should you wish to represent finer areas using SAS, you have the choice of purchasing specific map sets from SAS or other vendors, or creating your own SAS map data sets yourself (I’ll leave a description of this process for a future paper!) Map data sets provided by SAS can be “projected” or “unprojected”. Projected map data sets have had their longitude and latitude converted to X and Y coordinates and have been “flattened” so as to present well on paper. While you can (and sometimes have to) re-project map data sets, the BEST representations come from projected map data sets. (For example, SAS has figured out the best angle of projection to represent the entire United States, and has added Alaska and Hawaii to the map.) If you want to subset a particular state from the U.S. map, you must re-project the map to create and/or adjust the X and Y coordinates. Below follows the output of a proc contents and test print for the SAS US map (maps.us.)

**SAS-Supplied Map Data Sets**

**US Map Contents**

The CONTENTS Procedure

**Data Set Name:** MAPS.US

**Observations:** 1525

**Member Type:** DATA

**Variables:** 4

**Engine:** V8

**Indexes:** 0

**Created:** 8:50 Thursday, June 8, 2000

**Last Modified:** 8:50 Thursday, June 8, 2000

**Protection:** Compressed: NO

**Data Set Type:** Sorted: NO

**Label:** United States, reduced-projected:

Copyright(C) 2000 SAS Institute Inc.

-----Engine/Host Dependent Information-----

**Data Set Page Size:** 4096

**Number of Data Set Pages:** 9

**First Data Page:** 1

**Max Obs per Page:** 184

**Obs in First Data Page:** 115

**Number of Data Set Repairs:** 0

**File Name:** C:\Program Files\SAS Institute\SAS\V8\maps\us.sas7bdat

**Release Created:** 8.0202M0

**Host Created:** WIN_NT

-----Alphabetic List of Variables and Attributes-----

<table>
<thead>
<tr>
<th>#</th>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Pos</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>STATE</td>
<td>Num</td>
<td>5</td>
<td>5</td>
<td>State Segment Number</td>
</tr>
<tr>
<td>2</td>
<td>SEGMENT</td>
<td>Num</td>
<td>5</td>
<td>0</td>
<td>State FIPS Code</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>Num</td>
<td>6</td>
<td>10</td>
<td>X Coordinate</td>
</tr>
<tr>
<td>4</td>
<td>Y</td>
<td>Num</td>
<td>6</td>
<td>16</td>
<td>Y Coordinate</td>
</tr>
</tbody>
</table>

**US Map Test Print**

<table>
<thead>
<tr>
<th>STATE</th>
<th>SEGMENT</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.16175</td>
<td>-0.10044</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.12305</td>
<td>-0.10415</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.12296</td>
<td>-0.10478</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.12667</td>
<td>-0.11010</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0.12629</td>
<td>-0.11467</td>
</tr>
</tbody>
</table>

As you can see, this projected map data set contains many points represented by X and Y coordinates, associated with states and segments within states.

Some “map” data sets SAS provides contain names of areas, associated with geographical points designated by both X and Y coordinates matching “projected” map data sets, and longitude and latitude matching “unprojected” map data sets. Below follows a proc contents and test print from a map data set that provides center points for US states.
BUILD YOUR PRESENTATION DATA SET(S)

Note: for the purposes of this paper all analytic data elements in the example shown have been scrambled to protect data confidentiality.

The most important aspect to building a presentation data set to be used with a map is to correctly "link" items to the geographic areas being used. In the next example shown, I have used a format to associate values for the variables STATE and QINAME with an HTML reference, which can then be used in a data step or procedure later. This could easily be done in a data step with a permanent format being stored following the use of proc format cntlin, but for the purposes of demonstrating the linking process it is shown the "long" way below.

BUILDING A CUSTOM MAP

Although the SAS-supplied maps are adequate for most purposes as they stand, you may want to build a custom map. For example, you might want to present an analysis of variables based on counties in the state of Montana only, or you might want to label states on the US map. Below follow two examples: one is to create a county map of Montana, and the other is to "annotate" the US map or label states.

Note that all the examples following do not represent entire programs, but merely portions of programs. Please eMail the author for complete programs. Note also that proper indentation has been altered in order to fit into the paper format.
The next step is to write a macro to create data tables and/or graphics which can be associated with the formatted HTML references created above.

```sas
/* this creates the reports for each state and qiname */
ods listing close;
%macro runit(stcode,qiname,qilab,stname);
ods html body="&stcode.&qiname..htm" style=styles.sasweb;
data &stcode.&qiname;
  length qiname $ 5 qilabel $ 40;
  set dd.q4_comb (keep=st_code fac_itl name fac_city
               mcare_id aj_qiname ds_qiname na_qiname
               os_qiname ajstqiname ajnsqiname ossqiname
               osnsqiname
               rename=(aj_qiname=ADJUSTED
                       ds_qiname=DENOMNTR
                       na_qiname=NUMERATR
                       os_qiname=OBSERVED
                       ossqiname=obsstmn
                       osnsqiname=obsallmn
                       ajstqiname=stfacmn
                       ajnsqiname=allfacmn)
               where=(st_code="&stcode.");
  qiname="&qiname";
  qilabel="&qilab";
  /* relabel and label variables */
  label qiname='Quality Indicator Acronym'
  and so on . . .
  allfacmn='All State Adjusted Mean';
  format adjusted stfacmn allfacmn observed obsstmn
   obsallmn 7.1 denomntr numeratr comma7.;
  format fac_itl $char10. mcare_id $char6.;
  /* here's a bunch of call symputs to do the titles */
call symput('adjstmn',left(put(stfacmn,7.1)));
call symput('adjusmn',left(put(allfacmn,7.1)));
call symput('obsstmn',left(put(obsstmn,7.1)));
call symput('obsusmn',left(put(obsallmn,7.1)));
run;
proc print data=&stcode.&qiname label uniform;
  var name fac_city mcare_id denomntr numeratr observed adjusted;
  id fac_itl;
  title1 "&stname / &qilab QI NAME=&qiname - All Data Randomized";
  title2 "STATE MEAN (OBSERVED) = &obsstmn";
  title3 "SIX-STATE MEAN (OBSERVED) = &obsusmn";
  title4 "STATE MEAN (ADJUSTED) = &adjstmn";
  title5 "SIX-STATE MEAN (ADJUSTED) = &adjusmn";
  footnote1 "ALL DATA RANDOMIZED TO PROTECT CONFIDENTIALITY";
  run;
ods html close;
%mend;
%runit(CO,CADL1,CC: LATE-LOSS ADL WORSENING,COLORADO);%runit(CO,CDRG1,CC: PREV OF ANTIPSYCH DRUG USE,COLORADO);
%runit(CO,CINFX,CC: INFECTIONS PREVALENCE,COLORADO);
and so on . . .
%runit(WA,PWALX,PAC: IMPROVEMENT IN WALKING,WASHINGTON);
Example:

WASHINGTON / CC: LATE-LDSS ADL WORSENING QI NA
- All Data Randomized
STATE MEAN (OBSERVED) = 6.5
SIX-STATE MEAN (OBSERVED) = 6.6
STATE MEAN (ADJUSTED) = 6.3
SIX-STATE MEAN (ADJUSTED) = 6.4

<table>
<thead>
<tr>
<th>MIN System</th>
<th>Facility Operating Name</th>
<th>City Where Facility Is Located</th>
<th>Medicare Provider Number</th>
<th># of Residents after exclusion/resuspending covariates</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000000003</td>
<td>ALDERCREST HEALTH &amp; DEHL CENT</td>
<td>EDMONDS</td>
<td>502346</td>
<td>77</td>
</tr>
<tr>
<td>0000000004</td>
<td>ALDERWOOD MAJOR</td>
<td>SPOKANE</td>
<td>50257</td>
<td>57</td>
</tr>
</tbody>
</table>

Note: Graphic is cropped to save room in the paper.

Example:

/* this creates the graphic link (itself an HTML table) from each state to the reports for each state and quality indicator */

data dd.stlink;
  length linkme $ 80;
  set ee.stqi;
  linkme=put(state,statstat.);
  label QT_LABEL='Quality Indicator Description';
  STQI='Quality Indicator Acronym';
  QI_LABEL='Quality Indicator Acronym';
run;

%macro details(st,sttit);
  ods html body="&st..htm" path=odsout;
  title1 "Quality Indicators for &sttit";
  proc print data=dd.stlink (where=(state=&st)) noobs label uniform;
    var STQI QT_LABEL;
    format STQI $qinamef.;
    run;
  ods html close;
  run;
%mend details;

%details(1,Alabama);
%details(2,Alaska);
and so on . . .
%details(56,Wyoming);

Example:

<table>
<thead>
<tr>
<th>Quality Indicator Acronym</th>
<th>Quality Indicator Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CACL1</td>
<td>CC: LATE-LDSS ADL WORSENING</td>
</tr>
<tr>
<td>CCREG</td>
<td>CC: PREV OF ANTPSYCH DRUG USE</td>
</tr>
<tr>
<td>CINFX</td>
<td>CC: INFECTIONS PREVALENCE</td>
</tr>
<tr>
<td>CPUAI</td>
<td>CC: PAIN—INADEQUATE MANAGEMENT</td>
</tr>
<tr>
<td>CPUCU</td>
<td>CC: PREV OF PRESSURE ULCERS</td>
</tr>
<tr>
<td>CPRI</td>
<td>CC: PREV OF RESTRAINTS USED DAILY</td>
</tr>
<tr>
<td>CWGST</td>
<td>CC: WEIGHT LOSS PREVALENCE</td>
</tr>
<tr>
<td>EDELX</td>
<td>PAC: FAILURE TO IMPROVE/MANAGE DELIRIUM</td>
</tr>
<tr>
<td>PPAIX</td>
<td>PAC: INADEQUATE PAIN MANAGEMENT</td>
</tr>
<tr>
<td>PWALX</td>
<td>PAC: IMPROVEMENT IN WALKING</td>
</tr>
</tbody>
</table>
PUTTING IT ALL TOGETHER

The portion of code presented below creates a “clickable” HTML document or map of the United States. Note the use of the annotate data set created above to label the states with their postal abbreviations.

```sas
ods html body='maphead.htm' path=odsout;
proc gmap data=dd.stlink map=maps.us;
    id state;
    choro state / discrete html=linkme coutline=black nolegend annotate=mapanno;
    title1 'Quality Indicators by State';
run;
quit;
```

The idea is that one can click on any state on the map to get more information about that state. In this case, I have scrambled the confidential data that was actually produced for a contract so that the reports presented are not “real”. If you open up the HTML map in a browser and click on the state of Washington in the map, you see the HTML table titled “Quality Indicators for Washington”. If you then click on one of the items in the Quality Indicator Acronym in the HTML table, you then get a table similar to the cropped table on ADL worsening for Washington above. The number, content and format of HTML “layers” is entirely up to you, the programmer. Please note, however, that each “layer” may contain many html files; space and naming conventions may become an issue!

The resolution of the graphics shown above suffer when they are compressed into a size that fits into a paper format and printed in black and white, but the graphics and reports are quite striking in their full, interactive glory on the computer screen.

I will have sample maps and programs available on a laptop at my poster during “Meet the Presenters” session. Please note that the program portions presented above are incomplete and cannot be run “as is”.

WHAT IF I’M AFRAID OF MICE?

“Clickable” HTML documents, graphics or maps are very cool. However, there may be times when interactive HTML is not appropriate. For example, you may wish to publish your data in the form of a PDF document, or a data user may wish to manipulate the appearance of a graphic themselves without actually performing any SAS programming. You might want to insert SAS-created HTML documents or graphics into other forms of software such as Microsoft Word or Excel. SAS version 8 and the Output Delivery System allow for such desires with such output formats as JAVA, Active-X, XML, PDF, etc. as well as HTML, SAS listings, and data sets. Below follows four pie charts created using the SAS system on the same data using different devices. Portions of the code used to create each pie chart precede the images.

**STATIC HTML PIE CHART:**

```sas
options ps=50 ls=80 errorabend;
libname ee '.';
title1 'ESRDPAC Bundle Analysis';
run;
/* specify the directory to write HTML files to */
filename odsout '.';
ods listing close;
goptions reset=global;
goptions device=gif hsize=4 in vsize=4 in ftext=swiss;
ods html body='daypiel.htm' path=odsout;
proc gchart data=ee.daylevel (where=(proccat in ('EPOGEN','FERRELECIT','IRON_DEXTRAN','HIV','CALCIJEX', 'CARNITOR','VENOFER','VANCOMYCIN','ZEMPLAR','OTHER')));
pie proccat / sumvar=charge type=mean
    percent=outside value=outside;
title1 f=swissb h=1 "MEAN ANCILLARY COSTS";
run;
quit;
ods html close;
```

---

**POSTERS**

NESUG 15
JAVA PIE CHART:

JAVA graphics can be manipulated by right clicking on the image with the mouse. Rolling over portions of the graphics gives additional information as portrayed below. You can change the graph type, colors, etc. with a single click. Note: You MUST have SAS Version 8 installed on the displaying system or the JAVA capabilities will not work.

```sas
options reset=global;
options device=java hsize=4 in vsize=4 in ftext=swiss;
ods html body='javapie1.htm' path=odsout;
pie proccat / sumvar=charge type=mean
  percent=outside value=outside;
title1 f=swissb h=1 "MEAN ANCILLARY COSTS";
run;
quit;
ods html close;
```

ACTIVE-X PIE CHART:

Active-X graphics can also be manipulated by right clicking on the image with the mouse. The capabilities are slightly different from those of JAVA graphics. Note: You MUST have SAS Version 8 installed on the displaying system or the Active-X capabilities will not work.

```sas
options device=activex hsize=4 in vsize=4 in ftext=swiss htext=.5;
ods html body='actxpie1.htm' path=odsout;
pie proccat / sumvar=charge type=mean
  percent=outside value=outside;
title1 f=swissb h=1 "MEAN ANCILLARY COSTS";
run;
quit;
ods html close;
```

PDF PIE CHART:

SAS produced PDF graphics are not interactive in any way. There are, however, advantages to producing large volumes of graphics and/or tables using the SAS V8 PDF device, as you can manipulate bookmarks and styles to make accessing the different graphics very easy. It also makes it hard to “manipulate the data”, a very credible fear of web publishers whether using the SAS system or not!

```sas
options reset=global;
ods pdf file='c:\esrdpac\pdfpie1.pdf';
pie proccat / sumvar=charge type=mean
  percent=outside value=outside;
title1 f=swissb h=1 "MEAN ANCILLARY COSTS";
run;
quit;
ods pdf close;
```
AN IMPORTANT TIP REGARDING SAS HTML GRAPHICS

It should be obvious from the examples above that creating a “clickable” graphic system can generate a LOT of HTML files. There are additional files that the SAS system creates when producing HTML graphics (HTML tables and reports are NOT affected by this). The SAS device used when producing HTML graphics such as maps and charts is the GIF device. SAS generates a separate GIF file for EACH TYPE of graphic, numbering multiple graphics of the same type sequentially. These GIF files MUST be in the directory where the HTML files reside and are called from using a browser. In addition, if you run more than one program generating the same types of graphics in the same directory, the GIF files from the most recent program will overwrite the older GIF files, creating errors. There may be a way to specify the names of the GIF files in your SAS program but I haven’t figured it out yet. Therefore, if you are ignorant like me, it is a good idea to place presentation systems, once they are created to your satisfaction, in separate sub-directories!

CONCLUSION

SAS Version 8 provides many new, and some improved older tools, to produce some truly amazing results. Intranet or internet publishing in a variety of useful forms becomes easy using SAS Version 8’s Output Delivery System. In particular, attractive and useful interactive graphic presentation systems such as “clickable” maps can be created using SAS Version 8’s capability to produce custom HTML graphics. It is even possible, with some clever coding, to animate the graphics, all completely within the SAS system. Presentation of data using the SAS system has come a long way since the ubiquitous PROC PRINT output of the 70’s, carrying the SAS system into the new millennium in STYLE!

REFERENCES

SAS Online Documentation (PC SAS V612, PC SAS V8, AIX UNIX SAS V612, AIX UNIX SAS V8, RED HAT LINUX 6.2 SAS V8)

The Complete Guide to the SAS Output Delivery System, Versions 7-1 and 8

ACKNOWLEDGMENTS

The author wishes to acknowledge Ray Pass and Dana Rafiee, who inspired the “clickable” map with their tutorials on adding hot-link drill-down capability to HTML output and Web Publishing, and Patrick McGown for his illuminating paper on publishing Adobe PDF's from SAS/Graph Output.

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:

Louise Hadden
Abt Associates Inc.
55 Wheeler St.
Cambridge, MA 02138
Work Phone: 617-349-2385
Fax: 617-349-2675
Email: louise_hadden@abtassoc.com

KEYWORDS

SAS; UNIX; LINUX; ODS; HTML; ACTIVE X; PDF; GIF; JAVA; SAS GRAPH; MAPS