

Cool Maps with Custom Areas

Devi Sekar, GlaxoSmithKline, RTP, NC

ABSTRACT

SAS/GRAPH® software provides map coordinates for US state and county boundaries and also for several other countries. What if I want to draw my own region boundaries on the US map other than the state or city boundaries that SAS/GRAPH provides? To be able to work with SAS/GRAPH, I needed my region boundary coordinates in SAS traditional map datasets. How to create these map coordinate datasets? This paper has the answers! This paper walks you through the entire process from ground up - starting with converting non readable SAS map coordinates to generating wonderful drill down maps. This paper assumes working knowledge of SAS/GRAPH and ODS.

BACKGROUND

GlaxoSmithKline (GSK) pharmaceuticals Sales department divides the entire US geography into several units called footprints or territories for sales purpose. Each territory is managed by a single sales representative. These footprints are further rolled into larger areas called regions. We needed to draw these sales footprint boundaries or region boundaries on the US map and display several metrics on the map such as product market share or product sales volume. We were desperate to generate these maps using SAS/GRAPH, so that we could make use of the powerful features available in SAS/GRAPH and SAS ODS. Territory Alignments department in our company generates footprint boundary coordinates in .MIP and .MID file formats with MapInfo software. These were unusable directly by SAS/GRAPH. My major stumbling block was how to read these coordinates into SAS. After researching the SAS online documentation, I discovered that SAS can convert ESRI shape files (spatial data formats) .SHP files into SAS/GRAPH traditional map data sets. However, I needed yet another software tool that converts our .MIP and .MID files to ESRI shape files. After doing some searching on the web, I did find a tool that can do this job! Once the coordinates were imported into SAS/GRAPH map datasets, half the job was done!

STEPS PRESENTED IN THIS PAPER

1. Convert the map coordinates to ESRI shape files
2. Import ESRI shape files into SAS map datasets with MAPIMPORT procedure.
3. Project US Map correctly with GPROJECT procedure.
4. Create coordinates for larger areas using the GREMOVE procedure.
5. Annotate the region map.
6. View smaller area map for a larger area.
7. Sample drill down maps – using GMAP procedure and SAS ODS.

1. CONVERT MAP COORDINATES TO ESRI SHAPE FILES

Map coordinates available in my company were in .MIP and .MID format. I needed to convert them into ESRI shape file format. I was able accomplish this with the help of another software tool namely Geographic Translator produced by the company, Blue Marble Geographics.

Blue Marble Geographics software products easily allows users to change the coordinate systems of their vector-based map data to literally any known coordinate system, or any custom systems that can be defined. More information can be found in www.bluemarblegeo.com

Geographic Translator is a simple GUI tool and is very easy to work with. Below is a screen shot.

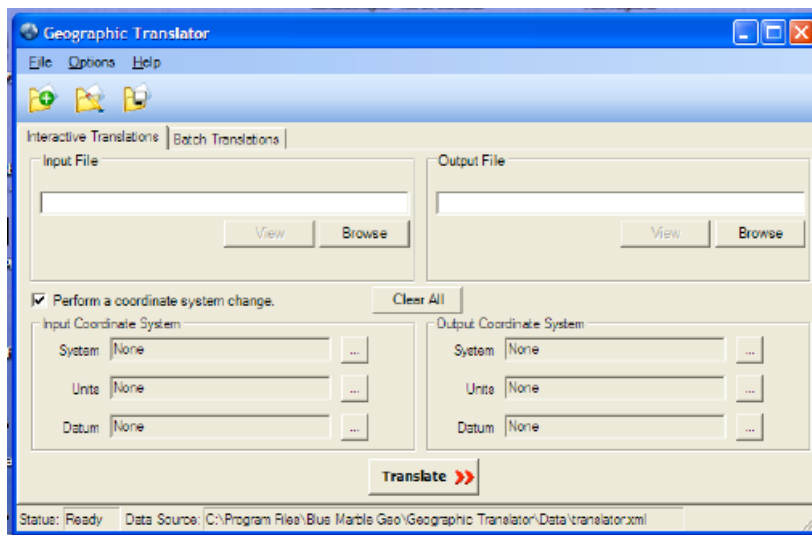


Figure 1.1

2. IMPORT ESRI SHAPE FILES INTO SAS MAP DATA SETS WITH MAPIMPORT PROCEDURE

With the above software, I have now converted my coordinates into ESRI shape files. Using the MAPIMPORT procedure in SAS/GRAPH, I can convert these ESRI shape files (spatial data formats) .SHP files into SAS/GRAPH traditional map data sets! A traditional map data set contains the X and Y coordinates that uniquely identify the areas in the map. For more information on this procedure, please refer to SAS Online Doc.

When you convert the MapInfo software files, .MIP and .MID format files to ESRI shape files, you will see a set of three files in your directory - .DBF file, .SHX file and .SHP file. Below is the description of each of these files.

File Extension	File Description
.dbf	identification information (field-identifier names and values) assigned to specific polygon(s)
.shx	shape information for the polygon(s) that compose the map. Note: These files are used with .shp files and cannot be imported by themselves.
.shp	combines the shape information for the polygon(s) that compose the map and the identification information (field-identifier names and values) assigned to the specific polygon(s)

In the PROC MAPIMPORT procedure, you will only mention .shp file. When you run this procedure, it looks for .dbf file and .shx file.

Example code for MAPIMPORT procedure is below:

```
proc mapimport out=FP_map datafile="C:\FPrt.shp";
    rename id=fprt;
run;
```

In the above example, FP_map dataset will have the X and Y coordinates usable by SAS/GRAPH. Here is a snapshot of the FP_map dataset:

	X	Y	SEGMENT	FPRT
1	-122.8086196	24.5723121	1	FP001A
2	-122.8053193	24.573816	1	FP001A
3	-122.7964387	24.5741478	1	FP001A
4	-122.7948274	24.5722116	1	FP001A
5	-122.7977752	24.5705109	1	FP001A
6	-122.7991027	24.5701539	1	FP001A
7	-122.8052734	24.5692761	1	FP001A
8	-122.8058137	24.5671782	1	FP001A
9	-122.8067932	24.5671473	1	FP001A
10	-122.8031632	24.5653107	1	FP001A
11	-122.8053193	24.573816	1	FP001A

Figure 2.1

Let us draw a simple choropleth map using this dataset:

```
title;
goptions reset=all;

proc gmap data=FP_map map=FP_map;
  id fppt;
  choro fppt;
run;
quit;
```

You can view the map below, with boundaries for each footprint.

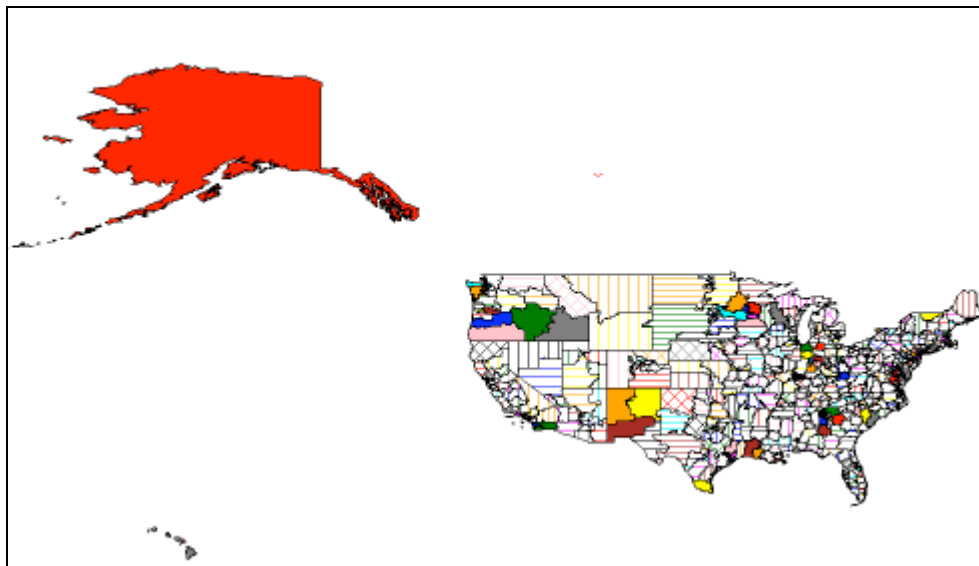


Figure 2.2

If you notice the above map closely, it looks stretched. This is because, the coordinates are not projected. Also, we want to change the position of Alaska and Hawaii to the bottom left. We can improve this map by adjusting Alaska and Hawaii coordinates and using the GPROJECT procedure.

3. CONVERT MAP COORDINATES WITH GPROJECT PRCEDURE

The GPROJECT procedure processes traditional map data sets by converting spherical coordinates (longitude and latitude) into Cartesian coordinates for use by the GMAP procedure. The process of converting coordinates from spherical to Cartesian is called **projecting**. For more information on this procedure, please refer to SAS Online Doc.

First, let us adjust the coordinates for Alaska and Hawaii Footprints, so that they are placed correctly on the map.

```
/* this part of code will move Alaska and Hawaii to better positions */
```

```
data FP_map_new;
  set fp_map;

  if fprt = 'FP001A' then do; /*Alaska */
    x = x*.3-70;
    y = y*.3+9;
  end;

  if fprt = 'FP072A' or fprt = 'FP073A' then do; /*Hawaii */
    x = x + 50;
    y = y + 7;
  end;
run;
```

Let us now project the map coordinates with GPROJECT procedure.

```
proc gproject data=FP_map_new out=gskmaps.fprt_map eastlong deg;
  id fprt;
run;
```

Let us draw the map and see how the map looks after we have applied the projections. Sure enough, this map (Figure 3.1) looks much better, the way we would want to see!!

```
title ;
options reset=all;
proc gmap data=gskmaps.fprt_map map=gskmaps.fprt_map;
  id fprt;
  choro fprt/nolegend;
run;
quit;
```

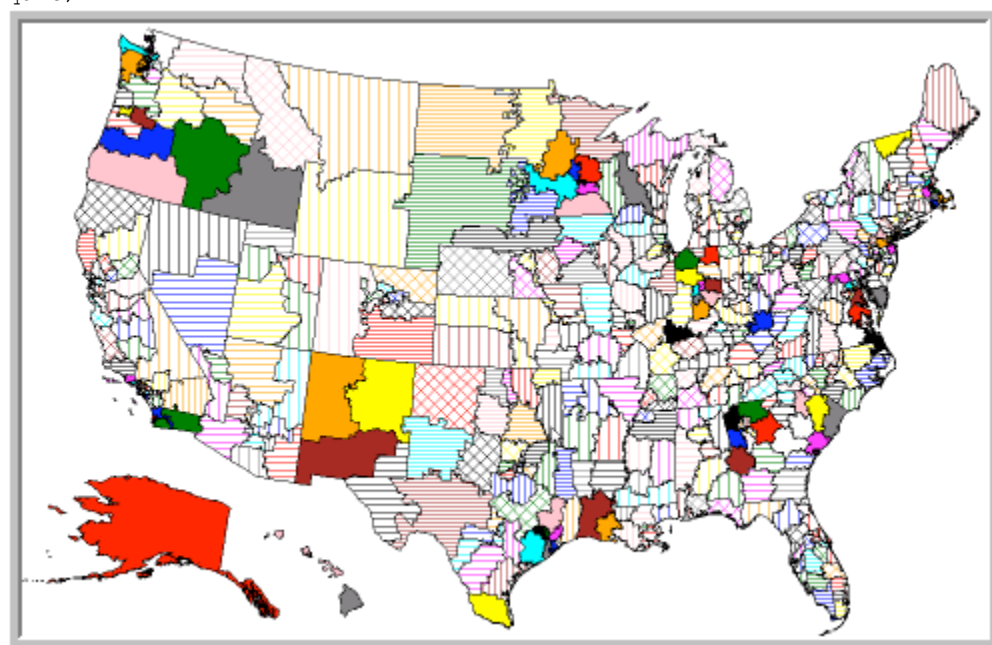


Figure 3.1

4. CREATE MAP COORDINATES FOR LARGER AREAS WITH GREMOVE PROCEDURE.

As I have mentioned earlier, these footprints are further rolled into larger areas called Regions. Having got the map datasets for the footprints, now I need map datasets for the larger areas, namely Regions. I was able to accomplish this with the help of the GREMOVE procedure. Let us see how the GREMOVE procedure works.

The GREMOVE procedure combines unit areas defined in a map data set into larger unit areas by removing shared borders between the original unit areas.

For the GREMOVE procedure to work, we need the footprint map dataset with its associated regions. At GSK, we do have a cross reference file that associates footprints to regions. By joining this cross reference file with the footprint map dataset, generated the dataset, **gskmaps.fprt_reg_map**, shown on the right.

	X	Y	SEGMENT	FPRT	REG
1	-122.8086196	24.5723121	1	FP001A	T2A
2	-122.8053193	24.573816	1	FP001A	T2A
3	-122.7964387	24.5741478	1	FP001A	T2A
4	-122.7948274	24.5722116	1	FP001A	T2A
5	-122.7977752	24.5705109	1	FP001A	T2A
6	-122.7991027	24.5701539	1	FP001A	T2A
7	-122.8052734	24.5692761	1	FP001A	T2A
8	-122.8058137	24.5671782	1	FP001A	T2A
9	-122.8067932	24.5671473	1	FP001A	T2A
10	-122.8031632	24.5653107	1	FP001A	T2A
11	-122.7952942	24.5673213	1	FP001A	T2A
12	-122.7951706	24.5643597	1	FP001A	T2A
13	-122.7866515	24.5685183	1	FP001A	T2A
14	-122.7850951	24.5657775	1	FP001A	T2A
15	-122.7842071	24.565464	1	FP001A	T2A
16	-122.788858	24.5620845	1	FP001A	T2A
17	-122.7855256	24.5605821	1	FP001A	T2A
18	-122.7865738	24.5590839	1	FP001A	T2A
19	-122.7862579	24.5580186	1	FP001A	T2A
20	-122.7888124	24.5566338	1	FP001A	T2A
21	-122.7886336	24.554289	1	FP001A	T2A

Figure 4.1

Using the above dataset, I was able to create a map datasets for region boundaries with GREMOVE procedure. For the GREMOVE procedure to work, the input dataset must have the x and y coordinates for the smaller area and the new variable that associates the smaller area to larger area. In the above dataset, x and y coordinates are for the footprint (smaller area) and the associated larger area variable: region.

The following code generates region level map dataset.

For the GREMOVE to work, the dataset needs to be sorted by the larger area.

```
proc sort data=gskmaps.fprt_reg_map out=fprt_reg_map;
  by reg;
run;
```

Now, apply the GREMOVE procedure. For this procedure to work correctly, it is necessary to have both the BY statement and ID statement in the GREMOVE procedure. The BY variables in the input map data set becomes the ID variables for the output map data set. For more information on this procedure, please refer to the GREMOVE procedure documentation in SAS online doc.

```
proc gremove data=fprt_reg_map out=gskmaps.reg_map;
  by reg;
  id fprt;
run;
```

We can now draw the region level map (Figure 4.2) using the region level map dataset.

```
proc gmap data=gskmaps.reg_map map=gskmaps.reg_map;
  id reg;
  choro reg/nolegend;
run;
quit;
```

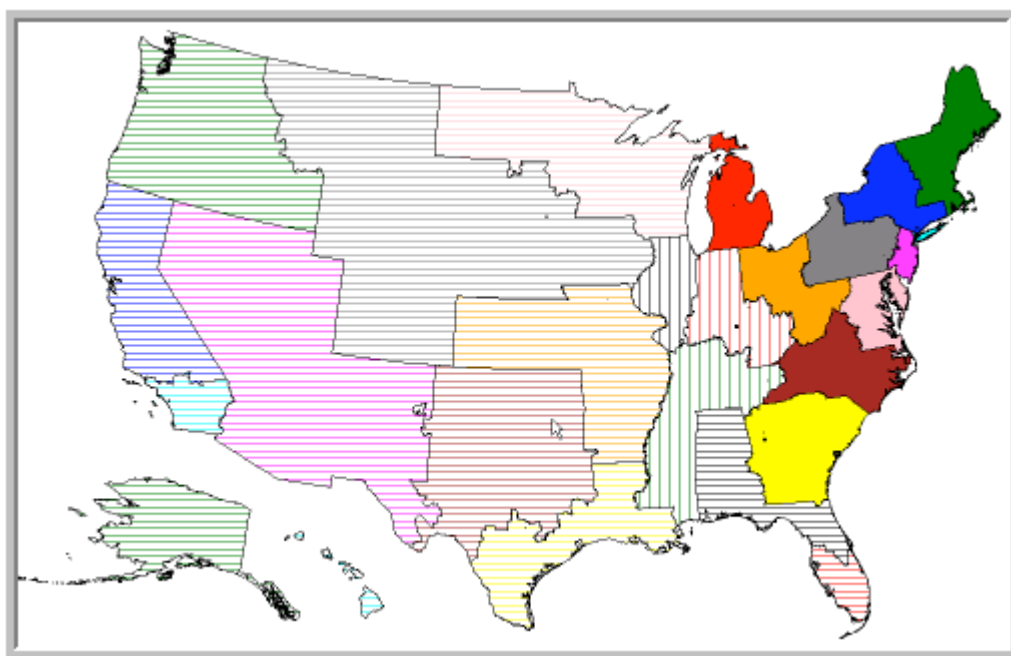


Figure 4.2

5. ANNOTATE THE REGION MAP

I want to label the regions in the above map now! For this purpose, I can make use of the ANNOTATE facility in SAS/GRAPH. SAS/GRAPH software provides several macros for annotating our graphs and maps. I was able to accomplish this easily, with the %maplabel macro, an annotation macro provided by SAS/GRAPH. This macro is provided in the %annomac library in SAS. You must submit the %annomac macro in a SAS session, before you can use any of the annotate macros. %annomac compiles annotate macros and makes them available for use.

Please refer to SAS ANNOTATE facility for more information on the annotate macros and annotate dataset variables.

I used the code below to generate annotate dataset for my map.

```
options mprint symbolgen;

proc freq data= gskmaps.reg_map noprint;
  tables reg/out=reg_names(drop=count percent);
run;

/* %annomac compiles annotate macros and makes them available for use*/
%annomac;

/* The maplabel macro expects the following parameters:
(map-dataset, attr-dataset, output-dataset, label-variable, id-list) */

%maplabel(%str(gskmaps.reg_map), reg_names, gskmaps.reg_anno, reg, reg);
```

You can now apply this annotate dataset to the region map. The code below generates a map with annotated regions, in Figure 5.1

```
title;
options reset=all;
proc gmap data=gskmaps.reg_map map=gskmaps.reg_map;
  id reg;
  choro reg/anno=gskmaps.reg_anno;
run;
quit;
```

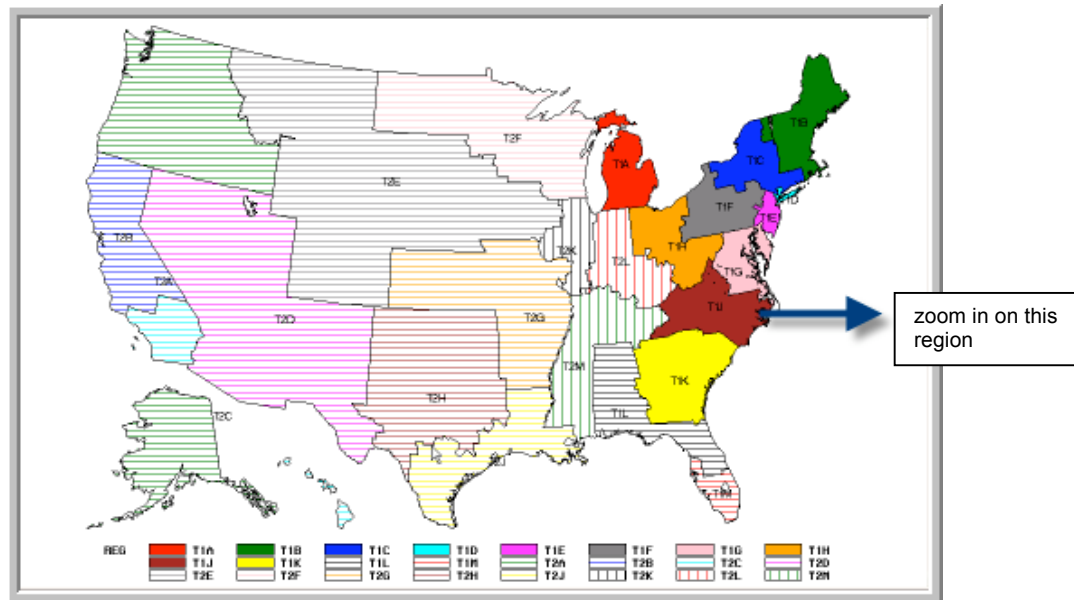


Figure 5.1

6. VIEW FOOTPRINT LEVEL MAP FOR A SPECIFIC REGION

Guess what! You can generate a footprint map associated with a specific region, by simply applying a where clause in the GMAP procedure! This is the coolest thing I have ever seen!! Sample code and the map are shown in 6.1 below, for T1J region (North and South Carolina region)

```
proc gmap
data=gskmaps.fprt_reg_map(where=(reg='T1J'))
    map=gskmaps.fprt_reg_map;
    id fprt;
    choro fprt/nolegend;
run;
quit;
```



Figure 6.1

7. PUT IT ALL TOGETHER

Sample Drill down maps:

I have applied all the techniques presented above and I was able to generate the following drill down maps. The maps below display region level US population count for the years 2000 and 2007. If you browse over each region, you can see the population count for that region. You can drill down on a specific region to view detailed region level map (Figure 7.2 below) with footprints level data! All this was accomplished using SAS ODS and HTML drill down facility in ODS.

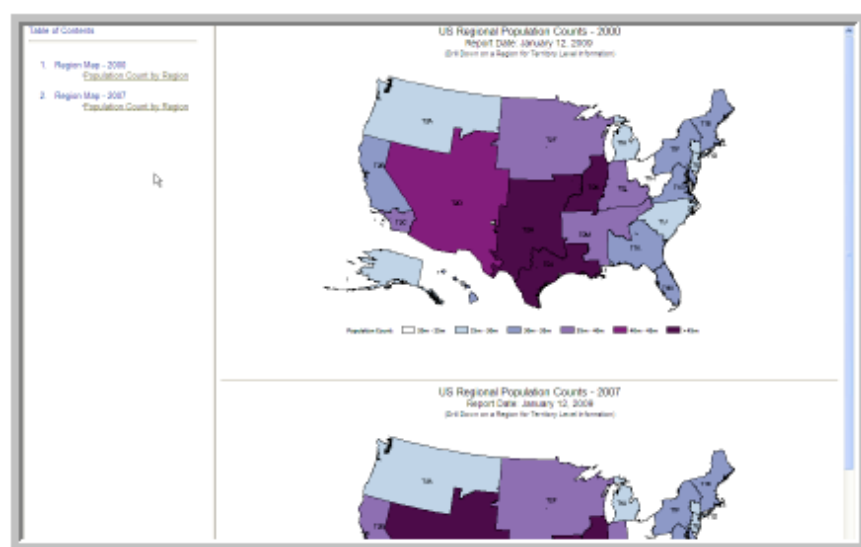


Figure7. 1

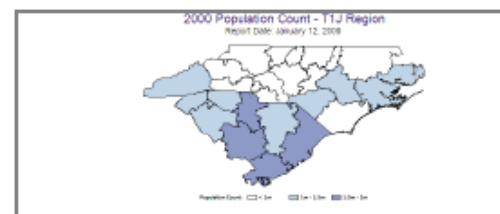


Figure7. 2

8. CODE BELOW GENERATES US MAP WITH REGION BOUNDARIES WITH DRILL DOWN CAPABILITY

8.1. SAS program for US Map with region boundaries and drill down capability

1. Data step below shows how to create the drill down variable, **my_link**. This variable will create a hot link in the map and lets you drill down to a region. Also, when you hover over a region you can see the population count and estimated growth for that region. Each region is associated with that region html file for each year. For example, for T1A region, you will need to have T1A_2000.html and T1A_2007.html files. Section 8.2 shows how to create these single region html files with footprint boundaries.

```
data reg_data_2000;
  set pop_reg_smry_2000;

/* Creates a tool-tip and drill down for the region */

  my_link = 'ALT="Region: ' || left(trim(reg)) ||
    '&#13; # Population Count: ' || left(trim(put(pop_cnt,comma15.))) ||
    '&#13; Estimated Growth ' || left(trim(put(est_growth,8.2))) ||
    '" href="' || trim(reg) || '_2000.html "' ;

run;
```

You can view a sample of the reg_data_2000 dataset below:

	reg	pop_cnt	est_growth	my_link
1	T1A	28566741	3.1626969741	ALT="Region: T1A # Population Count: 28,566,741  Estimated Growth 3.16 " href="T1A_2000.html "
2	T1B	31138601	3.4743767869	ALT="Region: T1B # Population Count: 31,138,601  Estimated Growth 3.47 " href="T1B_2000.html "
3	T1D	28622742	4.3272357162	ALT="Region: T1D # Population Count: 28,622,742  Estimated Growth 4.33 " href="T1D_2000.html "
4	T1E	25251475	3.9170481554	ALT="Region: T1E # Population Count: 25,251,475  Estimated Growth 3.92 " href="T1E_2000.html "
5	T1F	32555248	0.8461987363	ALT="Region: T1F # Population Count: 32,555,248  Estimated Growth 0.85 " href="T1F_2000.html "
6	T1G	32076215	9.3549918301	ALT="Region: T1G # Population Count: 32,076,215  Estimated Growth 9.35 " href="T1G_2000.html "
7	T1H	23166819	3.0620109148	ALT="Region: T1H # Population Count: 23,166,819  Estimated Growth 3.06 " href="T1H_2000.html "

Figure 8.1

2. Group population count into appropriate ranges.

```
proc format;
  value regpop
    20000000-25000000 = '20m - 25m'
    25000001-30000000 = '25m - 30m'
    30000001-35000000 = '30m - 35m'
    35000001-40000000 = '35m - 40m'
    40000001-45000000 = '40m - 45m'
    45000001-high      = '>45m';
run;
```

3. Code for generating the main maps.

```
filename odsout "C:\SAS Paper\Population maps";

ods listing close;

ods html path=odsout (url=none)
  frame='index.html'
  body="regmap.html"
  contents='toc.html'
  style=sasweb;

ods proclabel 'Region Map - 2000';

/* Region Maps */

title ;
goptions reset=all
device=gif
gsfmode=replace
xpixels=640
ypixels=480
ftext='Tahoma/bo'
;

/* RGB codes for variations for Violet color */

pattern1 v=ms c=white;      pattern2 v=ms c=cxbfd3e6; pattern3 v=ms c=cx8c96c6;
pattern4 v=ms c=cx8c6bb1; pattern5 v=ms c=cx810f7c; pattern6 v=ms c=cx4d004b;

/* Legend for the map */

legend1 label=(justify=c 'Population Count:');
```

```

/* Code below generates US map with region boundaries. html=my_link provides the drill
down capability. */

proc gmap data=reg_data_2000 map=gskmaps.reg_map;

    id reg;
    choro pop_cnt/discrete des='Population Count by Region'   coutline=black
                        anno=mydata.reg_anno html=my_link    legend=legend1;

    format pop_cnt regpop.;

    title1 font='Arial' height=15pt "US Regional Population Counts - 2000";
    title2 font='Arial' height=11pt "(Drill Down on a Region for Territory Level
    Information)";

run;
quit;

ods html close;

ods listing;

/* Clears the graphs in the catalog */

proc datasets library=work memtype=catalog;
    delete gseg;
run;
quit;

```

8.2 SAS Program for single region level map with footprint boundaries

When you drill down a region, it looks for the map related to specific region. Below is the code to create a region level map for each region with footprint boundaries. We need one html file for each region and for each year. Place it in a macro and loop through it for each region.

```

/* my_link variable below enables you to hover over each footprint and view the
population count and est. growth for that footprint */

data graphdata_2000;
    length fprt $6;
    set pop_data_2000;

    my_link = 'ALT="Footprint: ' || left(trim(fprt)) ||
              '&#13;Population Count: ' || left(trim(put(pop,comma8.))) ||
              '&#13;Estimated Growth ' || left(trim(put(est_growth,8.2))) || '";

run;

/* format population groups */

proc format;
    value popgrp
        30000-1000000    = ' < 1m'
        1000000-1500000  = '1m - 1.5m'
        1500001-2000000  = '1.5m - 2m'
        2000001-3000000  = '2m - 3m'
        3000001-high     = '>3m';
run;

/* put a list of regions into macro variables*/

proc sql noprint;
    select count(distinct reg) into: regcnt

```

```

    from graphdata_2000;

    select distinct reg into: reg1 - :reg%trim(&regcnt.)
    from graphdata_2000;
quit;

/* Generate two html files for each region -one for 2000 population and one for 2007
population */

%macro repeatme(year=);

filename odsout "C:\Documents and Settings\ds30144\My Documents\SAS Paper\Population
maps";

%do i= 1 %to &regcnt;

ods html path=odsout (url=none)
        body="&&reg&i.._&year...html"
        style=meadow;

goptions reset=all
device=gif
gsfmode=replace
gunit=pct
xpixels=480
ypixels=320
ftext='Tahoma/bo';

pattern1 v=ms c=white;      pattern2 v=ms c=cxbfd3e6;
pattern3 v=ms c=cx8c96c6;  pattern4 v=ms c=cx8c6bb1;

legend1 label=(height=3 justify=c 'Population Count:' )
        value=(height=3);

proc gmap data=graphdata_&year(where=(reg = "&&reg&i"))
        map=gskmaps.fprt_reg_map(where=(reg = "&&reg&i"));

    id fprt;
    format pop popgrp.;
    choro pop / discrete coutline=black html=my_link
        name="&&reg&i.._&year " des="&&reg&i.._&year " legend=legend1;
    title1 font='Arial' height=18pt color=vipb
        "&year Population Count - &&reg&i Region" ;

run;
quit;
title;
ods html close;

proc datasets library=work memtype=catalog;
    delete gseg;
run;
quit;

%end;

%mend;

%repeatme(year=2000); /* Repeat the above macro for year 2000 */

%repeatme(year=2007); /* Repeat the above macro for year 2007 */

ods listing;

```

CONCLUSION

SAS/GRAPH coupled with ODS is very powerful - one can develop sophisticated graphs and maps. Developing the building blocks for SAS/GRAPH was well worth the effort. I hope you get to make use of some of the techniques presented in this paper and benefit by them.

REFERENCES

SAS Online Doc 9.1.3 for the Web
SAS Institute, Cary NC

Specifying Colors in SAS/GRAPH Programs section, SAS Online Doc 9.1.3
SAS Institute, Cary NC

Code Samples and Technical Tips (<http://support.sas.com/sassamples/index.html>)

Blue Marble Geographic software: www.bluemarblegeo.com

Massengill Darrell 2005, "Tips and Tricks |||: getting the most from your SAS/GRAPH maps", SUGI 30 SAS Presents Handout and Example SAS Programs Download. SAS Institute

Massengill Darrell, 2005. "Tips and Tricks: Using SAS/GRAPH Effectively", SUGI 30 Proceedings, Philadelphia

ACKNOWLEDGEMENTS

I would like to thank Darrell Massengill of SAS Institute and Kriss Harris of GlaxoSmithKline for reviewing this paper and for their valuable suggestions.

CONTACT INFORMATION

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

Devi Sekar
GlaxoSmithKline
C1106, Five Moore Drive
RTP, NC 27709
(919) 483-4993
devi.k.sekar@gsk.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks of SAS Institute Inc., in the USA and other countries. ® indicates USA registration. Other brand and products names are trademarks of their respective companies.