ABSTRACT
This poster shows a display of flags depicting worldwide locations of SAS subsidiaries, regional offices and distributors. For many users, annotating specific locations on a SAS map data set has not been a simple task, primarily due to the fact that the data is not in latitude and longitude coordinates. Currently under development, is the effort to convert all map datasets into latitude and longitude. The programs used in this poster is an example of how one will be able to use a converted map dataset to annotate cities for example.

INTRODUCTION
Although this poster depicts only locations of SAS® offices and distributors, thus positioning the SAS® System in every continent; the countries displayed here are only a subset of the countries where one will actually find SAS users. For example, in 1994, in the Asia/Pacific Region alone, SAS was used in 48 different countries.

THE PROCESS
- preparing flags
- creating maps
- circle of rectangle with flags
- putting it all together

Flags
The flags used here, were imported into SAS, using the IMAGE procedure, and stored in a SAS catalog of type IMAGE. They were scaled in at 100X60 pixels.

The Map Program
%macro makemap(cont,cname,clause)
find the center of the continent to be used by the projection.

%inc 'center.sas';
%CENTERS(maps.worldmap,where=(cont=&cont));
subset the data using specified latitude and longitude. The IDENTIFY variable is used, temporarily, for separating the map data from the annotate data after the projection.

data sub;
  retain identify 'M';
  set maps.worldmap(where=(&clause));
run;
create a list of countries in continent subset

proc freq data=sub(where=(cont=&cont));
tables id/list out=cntlist(keep=id) noprprint;
run;
create the annotate dataset of site information subsetting only the ones for the current map subset; since the latitude and longitude values in the site dataset are in degrees, they are converted to radians using the constant 57.296.

%let el=%str(if w1 and w2 then output);
%let e2=%str(if w1 and w2 or l in (331 525 541 542 928) then output);
data paste;
  retain color 'black' xsys ysys '2'
  position 'B' identify 'P';
  keep function color where size xsys ysys x y text position country style id identify;
merge map.sasites(in=w1) cntlist(in=w2);
run;
combine the map and annotate data for projecting.

%inc 'hammer.sas';
%hammer(both,&cname,l,&phi,&lam);
project the data above using the Hammer projection technique; Phi and LAM macro variables are computed and stored by the centers programs described earlier; BOTH, the last parameter, is the input dataset; $CNAME, the 2nd parameter, represents the continent name; the 3rd parameter, (1), indicates that the input data is reversed — as is the case with the WORLDMAP data.

%inc 'hammer.sas';
%hammer(both,&cname,l,&phi,&lam);
separate the projected data back into the map and annotate datasets
data map(keep=country id segment x y)
anno(keep=function color when size text
position x y xsys ysys country
id style);
set &cname;
if identify='P' then output a:mo;
else output map;
run;

this step uses the annotate dataset to create a new variable in the
map dataset, that identifies countries where SAS has a distributor,
or subsidiary/regional office.

data map.&cname;
merge map paste(keep=id in-chk);
by id;
if chk the~ id2=1; else id2=2;
run;
display the map with cities annotated.
goptions nodisplay reset=pattern
colors=(yellow green);
proc gmap map=map.&cname data=map.&cname
gout=map.hammer;
id id id2; choro id2/discrete nolegend
coulime-gray name='&cname' annotate=anno
pattern v=s r=100;title; run;
%mend;

Macro Execution
%let cc=%str((x ge 52/57.29 and x le 165/57.29
and (y le 90/57.29 and y ge 15/57.29));
%makemap(91,namerica,&cc);

Output 1 North America

%let cc=%str((x ge 35/57.29 and x le 85/57.29
and (y le 15/57.29 and y ge -60/57.29));
%makemap(92,samerica,&cc);

Output 2 South America
%let cc=%str((x ge -100/57.29 and x le 80/57.29
and (y le 90/57.29 and y ge -20/57.29;
%makemap(93.europe,&cc);

Output 3 Europe
%let cc=%str((x ge -60/57.29 and x le 20/57.29
and (y le 38/57.29 and y ge -38/57.29;
%makemap(94.africa,&cc);

Output 4 Africa
%let cc=%str((x ge -160/57.29 and x le -20/57.29
and (y le 90/57.29 and y ge 0));
%makemap(95.asia,&cc);
proc freq data=paste:
   tables id/list outs=idlist noprint;
run;
data _null_
   set idlist end=last;
   if last then call symput('ids' ,_n_);
run;
data _null_
   length n 3;
n=ids;
mby=360/n;
call symput('mby' ,mby);
run;
%let cc=%str((x ge -180/57.296 and x Ie -90/57.296)
   and (y le 15/57.296 and y ge -60/57.296));
%makemap(96,spacific,&cc)
create and output the rectangles

goptions gunit=pct;
data cflag;
   length function color style 5 0;
   retain xsys ysys hsys '3';
do fig=0 to 330 by &mby;
   function='piecntr';x=50;y=50;size=45;
   output;
   function='piexy';size=1;angle=fig;
   output;
   function='cnt12txt';output;
   function='label';style='rect';text='r' ;angle=0; color='gray';size=15.4;x=.;y=.;
   output; end; run;
goptions cback=black;
proc ganno anno=cflag;
title; run;

to position the flags, after choosing edit graph:
   file -> open -> link -> to image
Putting it all Together
The GRSEG1S were exported to TIFF format, which were then converted to GEN files, to be used by the NVISION and PAINT procedures in SAS.
The maps were read into NVISION and rendered over a sphere. The output was stored as a GEN file.
The new map from NVISION and the flags, were both read into PAINT and pasted together. The final image was stored as a TIFF file.

SUMMARY

Although the Graphics Editor could have been used to "Put it all Together", I found that the method described above, greatly increased output resolution.

The process of positioning the flags within the rectangles, could probably be automated by creating a Frame and using SCL to place the images.

ACKNOWLEDGEMENTS

Many thanks to Enrico Leoni and the members of the Advanced Visualization department for their assistance with NVision and to Jesse Chavis and the Creative Services department for their assistance with the flags.

REFERENCES


Pearson, Frederick (1992), "Map Projection Equations", Naval Research Weapons Center, Dahlgren, VA. 147,313.


GENERAL REFERENCES

Flag Catalog, Alamo Flag Company (1993)

Flags of the World, Germany (1989)
APPENDIX

CENTER.SAS

%macro centers(dsn,code);
%global phi, lam;
/* S A S S A M P L E L I B R A R Y */
/* NAME: GMAPCTR */
/* TITLE: CALCULATE CENTERS OF GRAVITY */
/* PRODUCT: GRAPHiNG */
/* SYSTEM: MVS CMS VWS AGS PRIMOS VSE */
/* PROCES: GMAP */
/* DATA: */
/* KEYS: GMAP SUGGS */
/* SUPPORT: GRAPHICS STAFF UPDATE: 12NOV87 */
/* REF: A PROGRAMMER'S GEOMETRY BY */
/* MISC: THIS PROGRAM CALCULATES THE */
/* CENTERS OF GRAVITY, OR VISUAL */
/* CENTERS, FOR COUNTRIES IN WORLD- */
/* MAP. THIS CENTER WILL BE USED FOR*/
/* PROJECTING THE DATA. */
GOPTIONS NOCELL;

/*********************************************
*/
DATA CENTER(KEEP=ID FLAG XCG YCG);
RETAIN XCOM YCOM PTR 1 SAVPTR 1 XOLD YOLD XCG
YCG 0 ARESUM 0;
DO PTRV=1 TO NOBSV;
SET POINTS POINT=PTRV NOBS=NOBSV;
XCOM=XLAST; YCOM=YLAST;
XCG=0; YCG=0;
ARESUM=0;
FIRSTPTR=1;
PTR=SAVPTR;
ENDPTR=PTR + NPOINTER - 1;
DO PTRM=PTR TO ENDPTR;
SET YGPS POINT=PTRM NOBS=NOBSM;
IF FIRSTM THEN DO;
XOLD=X; YOLD=Y;
SAVPTR=PTRM + NPOINTS;
FIRSTM=0;
END;
ARETRI=(XCOM - X) * (YOLD - YCOM) +
(XOLD - XCOM) * (Y - YCOM);
XCG=XCG + ARETRI * (X + XOLD);
YCG=YCG + ARETRI * (Y + YOLD);
ARESUM=ARESUM + ARETRI;
XOLD=X; YOLD=Y;
END;
AREINV=1.0 / ARESUM;
XCG=('{XCG AREINV + XCOM} * 0.333333);
YCG=('{YCG * AREINV + YCOM} * 0.333333);
FLAG='C';
OUTPUT;
END;
STOP;
RUN;

data _null_; set center;
  phio=ycg*57.296;
  lamo=-(xcg*57.296);
  call symput('phi',phio);
  call symput('lam',lamo);
run;
%mend centers;

DATA CENTER(KEEP=ID FLAG XCG YCG);
RETAIN XCOM YCOM PTR 1 SAVPTR 1 XOLD YOLD XCG
YCG 0 ARESUM 0;
DO PTRV=1 TO NOBSV;
SET POINTS POINT=PTRV NOBS=NOBSV;
XCOM=XLAST; YCOM=YLAST;
XCG=0; YCG=0;
ARESUM=0;
FIRSTPTR=1;
PTR=SAVPTR;
ENDPTR=PTR + NPOINTER - 1;
DO PTRM=PTR TO ENDPTR;
SET YGPS POINT=PTRM NOBS=NOBSM;
IF FIRSTM THEN DO;
XOLD=X; YOLD=Y;
SAVPTR=PTRM + NPOINTS;
FIRSTM=0;
END;
ARETRI=(XCOM - X) * (YOLD - YCOM) +
(XOLD - XCOM) * (Y - YCOM);
XCG=XCG + ARETRI * (X + XOLD);
YCG=YCG + ARETRI * (Y + YOLD);
ARESUM=ARESUM + ARETRI;
XOLD=X; YOLD=Y;
END;
AREINV=1.0 / ARESUM;
XCG=('{XCG * AREINV + XCOM} * 0.333333);
YCG=('{YCG * AREINV + YCOM} * 0.333333);
FLAG='C';
OUTPUT;
END;
STOP;
RUN;

GOPTIONS NOCELL;

*********************************************;
** THIS DATA STEP CREATES TWO DATA SETS: MAP ;
** AND POINTS. THIS COUNTY POINTS FOR THE;
** SELECTED STATE ARE SUBSETTED OUT OF THE THE ;
** SAS DATA SET COUNTIES, AND THEN OUTPUTTED TO;
** MAP. THE LAST POINT AND THE TOTAL NUMBER OF ;
** POINTS USED TO CREATE EACH COUNTRY ARE OUT-;
** PUTTED TO POINTS. */
*********************************************;

DATA MAP(KEEP=XLAST YLAST NPOINTER);
  POINTS(KEEP=XLAST YLAST NPOINTER);
  SET &dsn(&code) end=last;
  FLAG='M' ;
  OUTPUT MAP;
  NPOINTER + 1;
  IF LAST THEN DO;
    XLAST=X; YLAST=Y;
    OUTPUT POINTS;
    NPOINTER=0;
  END;
RUN;

*********************************************;
** THIS DATA STEP CREATES A DATA SET CALLED ;
** CENTERS. THE CENTERS ; OF GRAVITY, XCG AND;
** YCG, ARE CALCULATED FOR EACH DATA SET. THE ;
** ALGORITHM USED WAS DERIVED FROM THE BOOK ;
** MENTIONED ABOVE (SEE THIS REFERENCE FOR ;
** DETAILS ON THE ALGORITHM). THE OBSERVATIONS;
** IN THE DATA SETS POINTS AND MAP ARE DIRECTLY;
** ACCESSED USING THE SET STATEMENT WITH THE ;
** POINT= AND NOBS= OPTIONS WITHIN DO LOOPS; ;
** (SEE SAS BASICS GUIDE FOR DETAILS). FOR ;
** EACH OBSERVATION IN POINTS (I.E. FOR EACH ;
** CENTER OF GRAVITY), INITIALIZE THE VARIABLES;
** THAT WILL BE USED TO CALCULATE XCG AND YCG ;
** THEN FOR EACH COUNTY (I.E. FOR ALL THE ;
** Observations ASSOCIATED WITH THE PARTICULAR ;
** COUNTY), USE THE INFORMATION TO CALCULATE ;
** XCG AND YCG. */
*********************************************;

DATA CENTER(KEEP=ID FLAG XCG YCG);
RETAIN XCOM YCOM PTR 1 SAVPTR 1 XOLD YOLD XCG
YCG 0 ARESUM 0;
DO PTRV=1 TO NOBSV;
SET POINTS POINT=PTRV NOBS=NOBSV;
XCOM=XLAST; YCOM=YLAST;
XCG=0; YCG=0;
ARESUM=0;
FIRSTPTR=1;
PTR=SAVPTR;
ENDPTR=PTR + NPOINTER - 1;
DO PTRM=PTR TO ENDPTR;
SET YGPS POINT=PTRM NOBS=NOBSM;
IF FIRSTM THEN DO;
XOLD=X; YOLD=Y;
SAVPTR=PTRM + NPOINTS;
FIRSTM=0;
END;
ARETRI=(XCOM - X) * (YOLD - YCOM) +
(XOLD - XCOM) * (Y - YCOM);
XCG=XCG + ARETRI * (X + XOLD);
YCG=YCG + ARETRI * (Y + YOLD);
ARESUM=ARESUM + ARETRI;
XOLD=X; YOLD=Y;
END;
AREINV=1.0 / ARESUM;
XCG=('{XCG * AREINV + XCOM} * 0.333333);
YCG=('{YCG * AREINV + YCOM} * 0.333333);
FLAG='C';
OUTPUT;
END;
STOP;
RUN;

DATA CENTER(KEEP=ID FLAG XCG YCG);
RETAIN XCOM YCOM PTR 1 SAVPTR 1 XOLD YOLD XCG
YCG 0 ARESUM 0;
DO PTRV=1 TO NOBSV;
SET POINTS POINT=PTRV NOBS=NOBSV;
XCOM=XLAST; YCOM=YLAST;
XCG=0; YCG=0;
ARESUM=0;
FIRSTPTR=1;
PTR=SAVPTR;
ENDPTR=PTR + NPOINTER - 1;
DO PTRM=PTR TO ENDPTR;
SET YGPS POINT=PTRM NOBS=NOBSM;
IF FIRSTM THEN DO;
XOLD=X; YOLD=Y;
SAVPTR=PTRM + NPOINTS;
FIRSTM=0;
END;
ARETRI=(XCOM - X) * (YOLD - YCOM) +
(XOLD - XCOM) * (Y - YCOM);
XCG=XCG + ARETRI * (X + XOLD);
YCG=YCG + ARETRI * (Y + YOLD);
ARESUM=ARESUM + ARETRI;
XOLD=X; YOLD=Y;
END;
AREINV=1.0 / ARESUM;
XCG=('{XCG * AREINV + XCOM} * 0.333333);
YCG=('{YCG * AREINV + YCOM} * 0.333333);
FLAG='C';
OUTPUT;
END;
STOP;
RUN;

GOPTIONS NOCELL;

*********************************************;
** THIS DATA STEP CREATES TWO DATA SETS: MAP ;
** AND POINTS. THIS COUNTY POINTS FOR THE;
** SELECTED STATE ARE SUBSETTED OUT OF THE THE ;
** SAS DATA SET COUNTIES, AND THEN OUTPUTTED TO;
** MAP. THE LAST POINT AND THE TOTAL NUMBER OF ;
** POINTS USED TO CREATE EACH COUNTRY ARE OUT-;
** PUTTED TO POINTS. */
*********************************************;

DATA MAP(KEEP=XLAST YLAST NPOINTER);
  POINTS(KEEP=XLAST YLAST NPOINTER);
  SET &dsn(&code) end=last;
  OUTPUT MAP;
  NPOINTER + 1;
  IF LAST THEN DO;
    XLAST=X; YLAST=Y;
    OUTPUT POINTS;
    NPOINTER=0;
  END;
RUN;

*********************************************;
** THIS DATA STEP CREATES A DATA SET CALLED ;
** CENTERS. THE CENTERS ; OF GRAVITY, XCG AND;
** YCG, ARE CALCULATED FOR EACH DATA SET. THE ;
** ALGORITHM USED WAS DERIVED FROM THE BOOK ;
** MENTIONED ABOVE (SEE THIS REFERENCE FOR ;
** DETAILS ON THE ALGORITHM). THE OBSERVATIONS;
** IN THE DATA SETS POINTS AND MAP ARE DIRECTLY;
** ACCESSED USING THE SET STATEMENT WITH THE ;
** POINT= AND NOBS= OPTIONS WITHIN DO LOOPS; ;
** (SEE SAS BASICS GUIDE FOR DETAILS). FOR ;
** EACH OBSERVATION IN POINTS (I.E. FOR EACH ;
** CENTER OF GRAVITY), INITIALIZE THE VARIABLES;
** THAT WILL BE USED TO CALCULATE XCG AND YCG ;
** THEN FOR EACH COUNTY (I.E. FOR ALL THE ;
** OBSERVATIONS ASSOCIATED WITH THE PARTICULAR ;
** COUNTY), USE THE INFORMATION TO CALCULATE ;
** XCG AND YCG. */
*********************************************;
HAMMER.SAS

%MACRO HAMMER(DSM, DSM2, REV, PHI0, LAM0);

/********************************************

NAME: HAMMER
TITLE: HAMMER PROJECTION CALC.
PRODUCT: GRAPH
SYSTEM: ALL
PROC:
DATA:
KEYS: CHAP SUGS
SUPPORT: LIZ SIMON UPDATE: 1994
REF: SEE TEXT BELOW

/* */

/********************************************

/* THIS PROGRAM WILL INPUT A MAP DATA SET AND*/
/* PROJECT IT USING THE HAMMER EQUAL AREA */
/* PROJECTION, ONE STANDARD PARALLEL. THE */
/* ALGORITHM WAS OBTAINED FROM THE FORTRAN */
/* PROGRAMS IN THE BOOK "MAP PROJECTION */
/* EQUATIONS" BY FREDERICK PEARSON OF THE */
/* NAVAL RESEARCH WEAPONS CENTER IN DAHLGREN,*/
/* VA. SEE PAGE 147 & 315 OF THAT REFERENCE. */

/* THE PROGRAM EXPECTS THESE INPUT MACROS:
/* DATA THE NAME OF THE INPUT SAS DATA SET*/
/* OTHERS THE LIST OF VARIABLES, OTHER THAN */
/* X AND Y, YOU WISH TO KEEP AFTER */
/* THE TRANSFORMATION. USUALLY, THESE*/
/* VARIABLES ARE THE ID VARIABLES & */
/* SEGMENT. */
/* REVERSE INDICATES WHETHER TO REVERSE X. IF*/
/* THE DATA ARE IN UNPROJECTED FORM, */
/* SAY: REVERSE LAMR=-LAMR */
/* */

/********************************************

%LET DATA::&DSM;
%IF &REV=1 %THEN
  %LET REVERSE=LAMR=-LAMR;
%ELSE %LET REVERSE=LAMR=LAMR;
DATA &DSM2;
  SET &DATA(RENAME=(X=LAMR Y=PHIR));
  RETAIN /* VALUE OF PI */
  PI 3.14159
  /* DEGREES-TO-RADIANS CONSTANT */
  RTD 57.296
  /* SEMI-MAJOR AXIS OF SPHEROID (WGS-72) */
  A 6378135
  /* ECCENTRICITY OF THE SPHEROID (WGS-72) */
  E .081819
  /* PHI0 IN RADIANS */
  PHIO
  /* LAM0 IN RADIANS */
  LAM0
  /* SCALING FACTOR */
  S 1
  R

/******INITIALIZE RETAINING CONSTANTS******/

IF _N_=1 THEN DO;
  R=A;
  PHIO=PHIO/RTD;
  LAM0=LAM0/RTD;
END;

/******RESET LAMR, SINCE INPUT IS REVERSED *****/
&REVERSE:
%PUT **&REVERSE**;

/*CREATE X AND Y FOR THE HAMMER PROJECTION*****/
LINK HAMMER:
RETURN:

/******HAMMER PROJECTION FORMULAE******/

HAMMER:

/***/
A1=SIN((LAMR-LAM0)/2);
A2=cos(phio)*tan(PHIR);
A3=sin(phio)*cos((lamr-lam0)/2);
IF (ABS(A2-A3) LE .0001) THEN
  ALF=SIGN(LAMR-LAM0)*PI/2;
ELSE

RHO=SQRT(2*(1-{R1+(R2*R3)));
X=2*R*S*RHO*SIN(ALF) ;
Y=R*S*RHO*COS(ALF)*SIGN(A2-A3);
RETURN;

%MEND HAMMER;

SITES DATA SET

Coordinates for US cities were taken from the USCITY dataset.
Most others were taken from the Cambridge World Gazetteer.
The coordinates are listed on a separate handout.