

SAS/GRAPH® Annotate Application

Sheila Fitzgerald Evans, SAS Institute Inc., Cary, N.C.

One useful application of the annotate facility in SAS/GRAPH® software is the ability to draw maps of city streets, lakes, creeks, railroads, and political and census boundaries from geographic base files. A DATA step (Appendix 1) creates a SAS® data set called STREETS from a tape available through Geographic Data Technology. Another DATA step (Appendix 2) is used to create an ANNOTATE= data set from the STREETS data set and a data set containing addresses. The GLSIDE procedure maps the streets and pinpoints specific addresses using the ANNOTATE= data set. Figure 6 is a sample application using input files of STREETS and ALARMS to create a custom map of fire calls for the city of Raleigh. The U.S. Census Bureau, through geographic base files called GBF/DIMES, has data available for most large metropolitan areas. Geographic Data Technology, Inc. has cleaved-up versions of the data available for a reasonable fee.

The DATA step that creates the SAS data set STREETS reads in a variety of information about the streets, lakes, creeks, railroads, and political and census boundaries from the Geographic Data Technology tape. These variable names are itemized and defined in the SAS DATA step in Appendix 1. The DATA step introduces a new variable called DRAWTYPE, which provides a method for SAS to locate the beginning and end of each street. For example, when the DATA step locates the first observation for a particular street name, DRAWTYPE is set to zero. Zero tells the MAPCITY macro (see below and Appendix 2) to generate the MOVE function for annotate. All other observations for that street name are set to one, telling the macro to generate the DRAW function.

The macro called MAPCITY creates an ANNOTATE= data set that maps streets and specific address locations if desired. In the macro, the ALARMS data set is merged with the larger STREETS data set to create a data set called MERGED. The ALARMS data set contains the street names or street names and address numbers you want to map. The STREETS data set contains the coordinates for all the streets and addresses in the city. (See code below.)

The macro also provides a method to plot other streets along with those specified in the ALARMS data set. This allows other streets to be used as reference points on the map. These streets are listed in an extra data set called REFERENCE. This data set is given to the macro in the last position of the parameter list as shown:

```
%MAPCITY(ALARMS, ANNOTATE, REFERENCE);
```

```

/*-----
| THIS SECTION OF CODE MERGES THE |
| ALARMS DATA SET AND THE REFERENCE |
| DATA, IF AVAILABLE, WITH THE |
| STREETS DATA SET. |
|-----*/
DATA MAP;
  RETAIN ID 1;
  SET &ALARMS &REFERENCE;
  KEEP NAME STYPE ID;
  RUN;

PROC SORT DATA=STREETS;
  BY NAME STYPE;
  RUN;

PROC SORT DATA=MAP;
  BY NAME STYPE;
  RUN;

```

```

/* KEEP ONLY THE STREETS LISTED */
/* IN THE DATA SET "MAP". */
DATA MERGED;
  MERGE MAP STREETS;
  BY NAME STYPE;
  IF ID=1;
  RUN;

```

The MAPCITY macro now creates the actual ANNOTATE= data set from the MERGED data set by adding the appropriate annotate functions. The macro stores the command to MOVE to the beginning of a new street or DRAW to the next observation on the current street. The commands to label streets are also stored.

```

/*-----
| DECLARE ANNOTATE VARIABLES. |
| DEFINE THE ANNOTATE REFERENCE |
| SYSTEM (5=WINDOW PERCENTAGE). |
|-----*/
DATA CITYMAP;
  %DCLANNO;
  %SYSTEM(5,5,5);
  SET MERGED;
  ARRAY XAXIS(2);
  ARRAY YAXIS(2);

/*-----
| SET CURRENT TO THE NEW |
| STREET BEING PROCESSED AND |
| STORE THE FIRST X,Y PAIR OF |
| EACH STREET. THE X,Y PAIR |
| IS USED IN DETERMINING THE |
| THE ANGLE TO PRINT THE |
| STREET'S NAME. |
|-----*/
IF ( CURRENT NOT=NAME ) OR
  ( _N_=1 )
THEN DO;
  CURRENT = NAME;
  XAXIS(1) = X;
  YAXIS(1) = Y;
  COUNT = 0;
  END;

/* COUNT THE OBSERVATIONS IN */
/* EACH STREET. */
COUNT=COUNT+1;

/*-----
| COUNT LOCATES THE SECOND |
| OBSERVATION OF EACH STREET. |
| X(Y)AXIS CONTAINS THE SECOND |
| OBSERVATION IN EACH STREET. |
|-----*/
IF COUNT=2
THEN DO;
  XAXIS(2)=X;
  YAXIS(2)=Y;

```

```

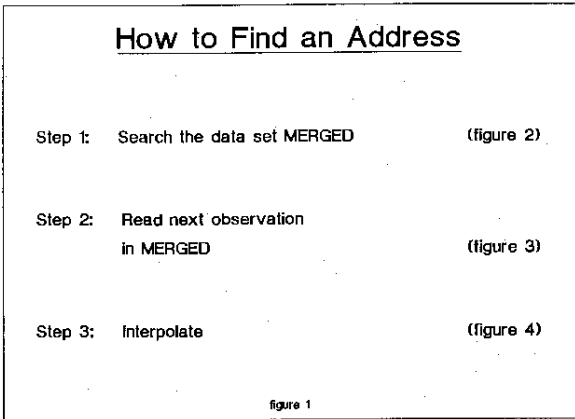
/*-----
| THE MACRO FINDANG DETERMINES |
| THE ANGLE TO PRINT THE      |
| STREET'S NAME. (APPENDIX 3) |
|-----*/
%FINDANG(XAXIS, YAXIS, ANGDEG);
/* PRINT THE STREET'S NAME. */
%LABEL(XAXIS(1),YAXIS(1),NAME,BLACK,
        ANGDEG,0,1.5,XSWISS,0);
END; /* COUNT */

/*-----
| THIS SECTION DRAWS THE STREETS. |
| DRAWTYPE IS FROM THE DATA SET |
| STREETS.                         |
|-----*/
IF DRAWTYPE=0 THEN FUNCTION='MOVE';
ELSE FUNCTION='DRAW';

OUTPUT;
RUN; /* CITYMAP */

```

If you wish to map addresses, a DATA step to find the coordinates of a specific address in the STREETS data set is executed within the macro. The data set created is called LOCATION. Figures 1, 2, 3, 4 and the code below show how the coordinates of a specific address are obtained.



Step 2

DATA MERGED:

OBS	DRAWTYPE	NAME	STYPE	X	Y	LowAdd	HAdd
1010	1	XXXX	ST	85.000	25.946	400	450
1011	0	XYZ	DR	0.000	25.917	3000	3023
1012	1	XYZ	DR	8.000	15.342	3000	3023
1013	1	XYZ	DR	22.222	96.207	3024	3047
1014	1	XYZ	DR	56.630	77.75	3048	3071
1015	1	XYZ	DR	56.630	77.75	3048	3071
1016	0	ZZZZ	AV	88.446	12.33	2	65

figure 3

Step 3

Interpolate Address:

$$dy = y_2 - y_1 = 77.750 - 96.207 = -18.457$$

$$dx = x_2 - x_1 = 56.630 - 22.222 = 34.408$$

$$ratio = (addnum - lowadd1) / (hiadd2 - lowadd1)$$

$$= (3034 - 3024) / (3071 - 3024) = 0.2127659$$

$$newx = x_1 + ratio * dx = 29.542849$$

$$newy = y_1 + ratio * dy = 92.27998$$

figure 4

Step 1

DATA MERGED:

OBS	DRAWTYPE	NAME	STYPE	X	Y	LowAdd	HAdd
1010	1	XXXX	ST	85.000	25.946	400	450
1011	0	XYZ	DR	0.000	25.917	3000	3023
1012	1	XYZ	DR	8.000	15.342	3000	3023
1013	1	XYZ	DR	22.222	96.207	3024	3047
1014	1	XYZ	DR	56.630	77.75	3048	3071
1015	1	XYZ	DR	56.630	77.75	3048	3071
1016	0	ZZZZ	AV	88.446	12.33	2	65

DATA ALARMS:

OBS	AddNum	NAME	STYPE
1	3034	XYZ	DR

figure 2

```

/*-----
| RENAME THE VARIABLES FOR      |
| COMPARISON.                  |
| SPD = STREET PREFIX DIRECTION |
| STYPE = STREET TYPE          |
| NAME = STREET NAME           |
|-----*/

DATA LOCATION;
SET ALARMS;
RENAME SPD = DIR;
RENAME STYPE= STY;
RENAME NAME = STREET;
RUN;

/*-----
| NOBS+NMNOBS IS EQUAL TO THE NUMBER|
| OF OBSERVATIONS IN THE MERGED     |
| DATA SET. N IS THE NUMBER OF    |
| OBSERVATIONS ON WHICH             |
| CALCULATIONS ARE BASED. NMNOBS   |
| IS THE NUMBER OF MISSING VALUES. |
|-----*/

PROC MEANS DATA=MERGED
  NOPRINT
  N
  NMNOBS;

VAR X;
OUTPUT OUT = STAT
  N = NOBS
  NMNOBS = NMNOBS;

RUN;

```

```

/*-----*/
| SYMPUT ASSIGNS N TO A MACRO |
| VARIABLE CALLED MAXOBS.  THUS, |
| YOU CAN PASS THIS VALUE INTO |
| ANY SAS DATA STEP. |
/*-----*/
DATA _NULL_;
  SET STAT;
  /* ADD THE NUMBER OF MISSING */
  /* OBSERVATIONS WITH THE NUMBER */
  /* OF OBSERVATIONS FOUND. */
  N=NOBS+NMOBS;
  CALL SYMPUT('MAXOBS',N);
  RUN;

/*-----*/
| LRIGHADD IS A CHARACTER VARIABLE. |
| TO SORT THE MERGED DATA |
| CORRECTLY, YOU MUST SORT BY A |
| NUMERIC VARIABLE (SORTBY) TO GET |
| PROPER NUMERIC ORDER. |
/*-----*/
DATA MERGED;
  SET MERGED;
  LENGTH SORTBY 6.;
  /* THE INPUT FUNCTION CONVERTS A */
  /* CHARACTER VALUE TO NUMERIC. */
  SORTBY=INPUT(LRIGHADD,5.);
  RUN;

PROC SORT DATA=MERGED;
  BY NAME STYPE SORTBY;
  RUN;

/*-----*/
| BELOW FINDS THE ADDRESS POINTS. |
/*-----*/
DATA LOCATION;
  ARRAY XNUM(2); /* CONTAINS THE X,Y */
  ARRAY YNUM(2); /* COORDINATES USED */
  /* IN INTERPOLATION.*/

  RETAIN;
  KEEP STREET STY DIR ADDRNUM NEWX NEWY;
  SET LOCATION;
  IDX = 1; /* INDEX POINTER. */
  FOUND = 0; /* 0=NOT FOUND. */
  MOREOBS = 1; /* LOOP VARIABLE. */

  DO WHILE (MOREOBS=1);
    SET MERGED POINT=IDX;

    /*-----*/
    | LOCATE THE SEGMENT POINT WHICH |
    | CONTAINS THE ADDRESS OF THE |
    | STREET CURRENTLY BEING PROCESSED. |
    /*-----*/
    IF ( NAME=STREET ) AND ( STYPE=STY )
      AND ( SPD=DIR ) AND
      ( ( ADDRNUM>=LRIGFADD ) AND
        ( ADDRNUM<=HLEFTADD ) )
    THEN DO;
      FOUND = 1;
      MOREOBS = 0;
      XNUM(1) = X;
      YNUM(1) = Y;
      LOWADD = INPUT(LRIGHADD,5.);

      /*-----*/
      | READ THE NEXT OBSERVATION IN |
      | THE DATA SET MERGED TO OBTAIN |

```

```

| A 2ND PAIR OF COORDINATES TO |
| BE USED FOR INTERPOLATION. |
/*-----*/
NEXTOBS=IDX + 1;
SET MERGED POINT=(NEXTOBS);
XNUM(2)=X;
YNUM(2)=Y;
HIADD=INPUT(HLEFTADD,5.);

/*-----*/
| INTERPOLATE ADDRESS POINT. |
/*-----*/
/* FIND THE CHANGE IN X AND Y. */
DY=YNUM(2)-YNUM(1);
DX=XNUM(2)-XNUM(1);
RATIO=((ADDRNUM-LOWADD) /
      (HIADD-LOWADD));
NEWX=XNUM(1)+RATIO*DX;
NEWY=YNUM(1)+RATIO*DY;
END; /* FOUND SEGMENT */

/*-----*/
| READ THE NEXT OBSERVATION AND |
| CHECK FOR END OF FILE. |
/*-----*/
ELSE DO;
  IDX=IDX + 1;
  IF IDX>MAXOBS THEN MOREOBS=0;
  END;
END; /* WHILE */

/*-----*/
| OUTPUT THE ADDRESS POINT. |
/*-----*/
IF FOUND=1
THEN OUTPUT;

ELSE PUT 'NOTE: Could not find address
        for' addrnum' 'street';
RUN; /* LOCATION */

```

After the LOCATION data set is created, the macro creates another ANNOTATE= data set, called ADDRESS, containing the commands to plot the address points from the data set LOCATION. The ADDRESS data set is then appended to the ANNOTATE= data set to create one data set called CITYMAP.

```

DATA ADDRESS;
  SET LOCATION;
  %DCLANNO;
  %SYSTEM(5,5,5);
  KEEP FUNCTION X Y XSYS YSYS HSYS WHEN
    POSITION SIZE STYLE COLOR TEXT;
  /* M=STAR */
  %LABEL(X,Y,'M',RED,0,0,5.0,SPECIAL,5);
  RUN;

DATA CITYMAP;
  SET CITYMAP ADDRESS;
  RUN;

```

The map is drawn within the program by executing PROC GSLIDE. The statement to invoke PROC GSLIDE is

```

PROC GSLIDE ANNOTATE=CITYMAP;
* RUN;

```

The X and Y coordinates are scaled to be in the range 0 to 100 (window percentage) instead of the default radian values (see Appendices 2 and 3). This flexibility makes annotate more useful because the map is easily customized.

The data used in this application is obtained through Geographic Data Technology, Inc. As part of the 1970 census, the Census Bureau used Dual Independent Map Encoding (DIME) to make computer-readable maps of 196 metropolitan statistical areas.

Many types of data can be combined through various geographic codes such as:

- Street addresses
- ZIP codes
- Civil division code
- Census county code
- Street intersections
- Census tracts
- Grid coordinates

This system is designed to help city managers, planners, administrators, and other decision makers extract and analyze data in the form of a city map. Two examples have been developed with the SAS/GRAPH annotate facility using the macro called MAPCITY. These examples use street addresses and ZIP code areas to describe pertinent information.

The map in figure 5 shows the number of fire calls in the Raleigh, North Carolina, area during a weekend in August, 1985. The fire calls are shown according to the ZIP code area where they occurred. The map indicates more of the fire calls (21.9%) occurred in ZIP code area 27601 — the downtown section.

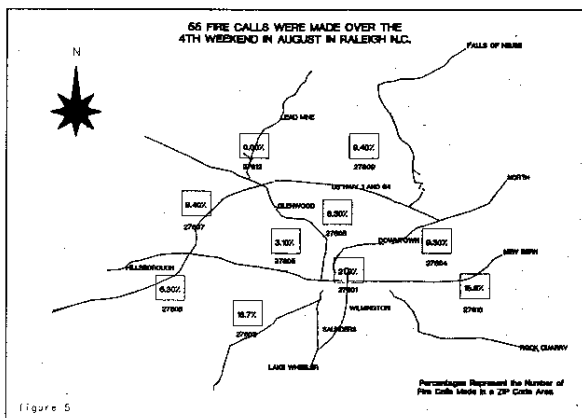
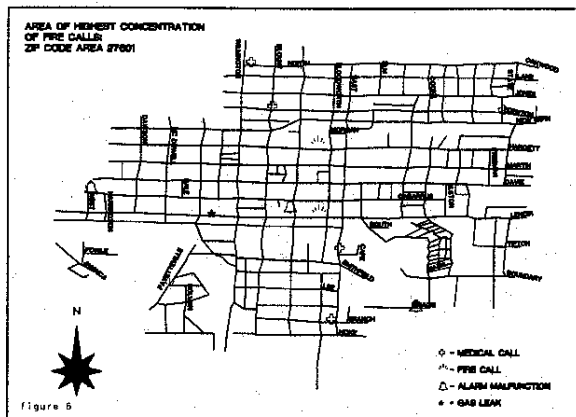


Figure 6 enlarges the map to include major streets in ZIP code area 27601. These streets are printed using the macro, and only the major streets are labeled to increase the clarity of the map. The symbols are created using the SAS/GRAPH type styles for special characters contained in Appendix 1 of the SAS/GRAPH® User's Guide, Version 5 Edition.



Street map tapes may be obtained from:

Geographic Data Technology, Inc.
13 Dartmouth College Highway
Lyme, NH 03768
Telephone: (603) 795-2183

SAS and SAS/GRAPH are registered trademarks of SAS Institute Inc., Cary, N.C., USA.

Appendix 1

```

-----*/
/* THIS DATA STEP READS IN THE DATA FROM THE TAPE GDTRAL. */
/* THE TAPE GDTRAL CONTAINS THE DATA NEEDED TO MAP THE STREETS */
/* OF RALEIGH ( OR OTHER CITIES ). */
-----*/

```

```

DATA STREETS;
  INFILE GDTRAL; /* GDT TAPE FOR RALEIGH */

INPUT @1 SPD $2. /* STREET PREFIX DIRECTION */
      @3 NAME $20. /* STREET, NON-STREET FEATURE NAME */
      @23 STYPE $4. /* STREET TYPE */
      @27 SSD $2. /* STREET SUFFIX DIRECTION */
      @29 FEATURE $1. /* FEATURE CODE */
      @30 ENLEFT $6. /* ENUMERATION DISTRICT LEFT */
      @36 ENRIGHT $6. /* ENUMERATION DISTRICT RIGHT */
      @42 FILLER $3. /* FILLER */
      @45 STCLASS $1. /* STREET CLASSIFICATION */
      @46 FROMAPN 3. /* FROM MAP (BASIC NUMBER) */
      @49 FROMAPS $2. /* FROM MAP (SUFFIX NUMBER) */
      @51 TOMAPN 3. /* TO MAP (BASIC NUMBER) */
      @54 TOMAPS $2. /* TO MAP (SUFFIX NUMBER) */
      @56 LIMFLAG $1. /* CODING LIMIT FLAG */
      @57 LLEFTADD $6. /* LEFT LOW ADDRESS */
      @63 HLEFTADD $6. /* LEFT HIGH ADDRESS */
      @69 LRHADD $6. /* RIGHT LOW ADDRESS */
      @75 RHIGHADD $6. /* RIGHT HIGH ADDRESS */
      @81 SMSA 4. /* STANDARD METRO STATISTICAL AREAS*/
      @85 RECNUM 6. /* RECORD CODE */
      @91 CKDIGIT 1. /* CHECK DIGIT */
      @92 CENTRANK 1. /* CENSUS TRACK (BASIC) */
      @96 CENTRANK 2. /* CENSUS TRACK (SUFFIX) */
      @98 CENRIGHB 4. /* CENSUS TRACK RIGHT (BASIC) */
      @102 CENRIGHS 2. /* CENSUS TRACK RIGHT (SUFFIX) */
      @104 ZIP 5. /* CURRENT ZIP CODE LEFT */
      @109 ZYRIGHT 5. /* CURRENT ZIP CODE RIGHT */
      @114 SMSACODE 4. /* SMSA CODE */
      @118 EXTFLAG 1. /* EXTEND FLAG */
      @119 FILLER2 $4. /* FILLER */
      @123 FROMNODE 4. /* FROM NODE */
      @127 TONODE 4. /* TO NODE */
      @131 CODE 4. /* PLACE CODE LEFT */
      @135 CODERIGH 4. /* PLACE CODE RIGHT */
      @139 STATELEF 2. /* FIPS STATE CODE LEFT */
      @141 COUNTYLE 3. /* FIPS COUNTY CODE LEFT */
      @144 MCD_COD 3. /* CIVIL DIV CODE / CENSUS COUNTY */
      @147 CUSZONEL 5. /* CUSTOM ZONE LEFT */
      @152 BLOCK 3. /* BLOCK LEFT */
      @155 FILLER3 $3. /* FILLER */
      @158 STATERIG 2. /* FIPS STATE CODE RIGHT */
      @160 COUNTYRI 3. /* FIPS COUNTY CODE RIGHT */
      @163 CIVDIV 3. /* MINOR CIVIL DIVISION CODE */
      @166 CUSZONER 5. /* CUSTOM ZONE RIGHT */
      @171 BLKRIGHT 3. /* BLOCK RIGHT */
      @174 FILLER4 $3. /* FILLER */
      @177 FROMSTAT 2. /* FROM STATE PLANE CODE */
      @179 TOSTATE 2. /* TO STATE PLANE CODE */
      @181 FILLERS $19. /* FILLER */
      @200 NETWORK 5. /* NETWORK ZONE */
      @205 Y 6.4 /* FROM LATITUDE */
      @211 X 7.4 /* FROM LONGITUDE */
      @218 Y2 6.4 /* TO LATITUDE */
      @224 X2 7.4 /* TO LONGITUDE */
      @231 Y3 7. /* FROM STATE PLANE */
      @236 X3 7. /* FROM STATE PLANE */
      @245 Y4 7. /* TO STATE PLANE */
      @252 X4 7. /* TO STATE PLANE */
      @259 EXLLOADD 10. /* EXPANDED LEFT LOW ADDRESS */
      @269 EXLHIADD 10. /* EXPANDED LEFT HIGH ADDRESS */
      @279 EXRLOADD 10. /* EXPANDED RIGHT LOW ADDRESS */
      @289 EXRHIADD 10. /* EXPANDED RIGHT HIGH ADDRESS */
      @298 FILLER6 $2.; /* FILLER */

```

OUTPUT;

```

-----*/
/* THE FOLLOWING DATA STEP CONVERTS THE LONGITUDE (Y, Y1) AND */
/* LATITUDE (X, X1) VALUES INTO RADIANS AND INTRODUCES THE */
/* VARIABLES LAT AND LONG. */
-----*/

```

```

DATA STREETS;
  SET STREETS;
  RETAIN D2R 0.0174533; /* USED TO CONVERT DEGREES INTO */
  LAT=X; /* RADIANS. (PI/180) */

```

```

LONG=Y;
LABEL LAT='LATITUDE IN DEGREES';
LABEL LONG='LONGITUDE IN DEGREES';
X=X*D2R; /* CHANGE X AND Y FROM DEGREE */
Y=Y*D2R; /* TO RADIANS. */
Y2=Y2*D2R;
X2=X2*D2R;
RUN;

```

```

/* SORT THE DATA SET BY THE NAME, STYPE AND THE ZIP CODE. */
PROC SORT DATA=STREETS;
  BY NAME STYPE ZIP;
RUN;

```

```

-----*/
/* PUT DATA SET INTO NEW FORM (BY ELIMINATING THE X2,Y2 VARIABLE */
/* AND OTHER UNNEEDED VARIABLES). */
-----*/

```

```

DATA STREETS;

  SET STREETS;
  BY NAME STYPE NOTSORTED;
  COUNTY=COUNTYLE;
  STATE=STFIPS('NC');
  RETAIN XLAST 0 YLAST 0;
  DROP XLAST YLAST;

```

```

/* BELOW DROPS UNNEEDED VARIABLES FROM THE DATA SET. */
DROP X2 Y2 X3 Y3 X4 Y4 CKDIGIT RECNUM;
DROP LIMFLAG TONODE FROMNODE TOSTATE FROMSTAT TOMAPN FROMAPN;
DROP COUNTYLE COUNTYRI STATELEF STATERIG;
DROP FILLER FILLER2 FILLER3 FILLER4 FILLER5 FILLER6;
DROP EXLLOADD EXLHIADD EXRLOADD EXRHIADD EXTFLAG CUSZONEL;
DROP CUSZONER NETWORK ENLEFT ENRIGHT STCLASS FROMAPS TOMAPS;
DROP SMSACODE D2R ZIPRIGHT;
DROP CODERIGH SMSA CIVDIV CENRIGHB CENRIGHS BLKRIGHT;

```

```

/* BELOW SETS CERTAIN VARIABLES TO A NEW LENGTH. */
LENGTH X Y ZIP MCD_COD CODE BLOCK CENTRANK CENTRANKS 4.;
LENGTH COUNTY STATE 2.;

```

```

-----*/
/* BELOW INTRODUCES A NEW VARIABLE CALLED DRAWTYPE. THIS VARIABLE */
/* IS USED TO MAKE DRAWING THE STREETS EASIER WHEN USING ANNOTATE. */
/* 0=THE FUNCTION 'MOVE' AND 1=THE FUNCTION 'DRAW'. */
/* THIS SECTION ALSO ELIMINATES THE X2,Y2 PAIR BY STORING THEM */
/* IN THE X AND Y VARIABLES. BEFORE THIS STEP, THE DATA IS IN */
/* THE FOLLOWING FORMAT: */
/* 1 ... X Y X2 Y2 ... */
/* THIS DATA STEP WILL CHANGE THE ABOVE FORMAT TO: */
/* 1 ... X Y ... */
/* 2 ... X Y ... <- X=X2, Y=Y2 */
-----*/
LENGTH DRAWTYPE 2.; /* 0=DISCONTINUITY 1=CONTINUE DRAWING */

```

```

IF XLAST NOT=X OR YLAST NOT=Y /* TEST UNPROJECTED VALUES. */
  THEN DO;
  DRAWTYPE=0;
  X=-X; /* RE-ORIENT THE VALUE. WHY: LONGITUDE INCREASES*/
  OUTPUT; /* FROM EAST TO WEST BUT ANNOTATE COORDINATES */
  END; /* INCREASE FROM LEFT TO RIGHT. TO MAKE EAST */
/* BE ON THE RIGHT INSTEAD OF THE LEFT EDGE OF */
/* THE SCREEN, YOU MUST REVERSE THE X AXIS, */
/* WHICH X=-X DOES. */

DRAWTYPE=1;
XLAST=X2; /* UPDATE WITH THE UNPROJECTED VALUE. */
YLAST=Y2;
X=-X2; /* RE-ORIENT THE VALUE. */
Y=Y2;
OUTPUT;

```

RUN;

Appendix 2

OPTICK CAPS;

```
/* THE CODE FROM APPENDIX 3 MUST BE INCLUDED BEFORE THIS MACRO. */
%INCLUDE IN(APPENDIX3);
```

```
%MACRO MAPCITY(INDATA,OUTDATA,MAPADD,FANCY,EXTRADS);
```

```

/*****
/*
/* PURPOSE:
/* THIS MACRO CREATES AN ANNOTATE DATA SET WHICH MAPS STREETS
/* LAKES, CREEKS, RAILROADS AND POLITICAL OR CENSUS BOUNDARIES.
/* IF 'MAPADD' EQUALS 'YES', THEN ADDRESSES WILL BE MAPPED BY
/* STARS. (SEE BELOW FOR DETAILS)
/*
/* INPUT:
/* THE INDATA DATA SET MUST CONTAIN AT LEAST THE NAME OF THE
/* STREET, ITS STREET TYPE (I.E. AV, ST) AND AN INTEGER VARIABLE
/* CALLED PRINTIT (1=PRINT THE NAME OF THE STREET ON THE GRAPH.
/* OTHERWISE THE NAME IS NOT PRINTED). THE VARIABLE FOR THE
/* STREET NAME SHOULD BE CALLED NAME, AND THE STREET TYPE SHOULD
/* BE CALLED STYPE. BOTH SHOULD BE TYPE CHARACTER. IN ORDER
/* TO MAP SPECIFIC ADDRESS LOCATIONS, YOU MUST SPECIFY ADDRESS
/* NUMBERS ALONG WITH THE NAME OF THE STREET, THE STREET TYPE
/* AND THE PRINTIT VALUE. THE VARIABLE NAME FOR THE ADDRESS
/* NUMBER SHOULD BE CALLED ADDRNUM, AND IT IS ALSO TYPE CHARACTER.
/*
/* DATA EXAMPLE;
/* INPUT ADDRNUM $ NAME $ STYPE $ EVENT $ PRINTIT;
/* CARD;
/* 3010 ABC CRK MEDICAL 1
/* 301 XYC CT FIRE 0
/* ;
/* RUN;
/*
/* %MAPCITY(EXAMPLE,ANNODATA,YES,NO);
/*
/* RESULT: A NEW DATA SET CALLED ANNODATA IS CREATED.
/* THE GRAPH WILL MARK THE ADDRESSES BECAUSE
/* THE PARAMETER 'MAPADD'='YES'. ONLY ABC
/* CRK'S NAME WILL BE PRINTED BECAUSE XYC'S
/* PRINTIT VARIABLE=0. SINCE 'FANCY'='NO',
/* ABC CRK'S NAME IS PRINTED BESIDE THE STREET.
/* (SEE PARAMETER SECTION FOR DETAILS)
/*
/* OUTPUT:
/* THE OUTPUT FOR THIS MACRO IS AN ANNOTATE DATA SET: 'OUTDATA'.
/* 'OUTDATA' IS THE NAME YOU WISH TO CALL THE ANNOTATE DATA SET.
/* A PROC GSLIDE ANNO='OUTDATA' OUTPUTS THE MAP.
/*
/* PARAMETERS:
/* 'INDATA' - THE INPUT DATA SET CONTAINING STREET NAMES,
/* STREET TYPE (ST, AV), THE PRINTIT VARIABLE
/* AND ADDRESS NUMBERS (OPTIONAL).
/* 'OUTDATA' - AN ANNOTATE DATA SET IS CREATED FROM THE 'INDATA'
/* DATA SET AND THE EXTRA DATA SET IF ONE IS
/* AVAILABLE.
/* 'MAPADD' - IF MAPADD='Y', 'YE' OR 'YES', THEN ADDRESSES
/* ARE MAPPED BY STARS. OTHERWISE ONLY THE STREETS
/* ARE DRAWN.
/* 'FANCY' - IF FANCY='Y', 'YE' OR 'YES', THEN THE NAMES OF
/* THE STREETS ARE PRINTED IN THE SAME DIRECTION
/* THE STREET FLOWS. OTHERWISE THE NAME IS PRINTED
/* BESIDE THE STREET.
/* 'EXTRADS' - THIS DATA SET IS AN EXTRA DATA SET CONTAINING
/* THE NAMES OF THE STREETS YOU WISH TO MAP ALONG
/* WITH THE 'INDATA' DATA SET.
/*
/* NOTES:
/* 1. IF X OR Y IS OUT OF RANGE (X<0 OR X>100, Y<0 OR Y>100),
/* THEN CHANGE THE NUMBERS IN THE SCALE SECTION BELOW.
/* 2. THIS PROGRAM USES ANNOTATE MACROS. SEE PAGES 153-155 IN
/* THE 'SAS/GRAPH USER'S GUIDE, VERSION 5 EDITION'.
/* 3. IF YOUR OUTPUT LOOKS INCORRECT, CHECK FOR AN OBSERVATION
/* WITH AN X OR Y THAT IS OUT OF RANGE (<0 OR >100). USUALLY
/* THE OBSERVATIONS FOLLOWING AN OBSERVATION CONTAINING AN
/* OUT OF RANGE VALUE DO NOT APPEAR ON THE GRAPH.
/*
/*****

```

```

| BELOW MERGES THE 'INDATA' AND THE 'EXTRADS' IF AVAILABLE WITH THE
| STREETS DATA SET.

```

```

| THIS NEW MERGED DATA SET IS USED TO CREATE THE ANNOTATE DATA SET
| CALLED 'OUTDATA'.

```

DATA STREETS;

```

RETAIN ID 1;
SET &INDATA &EXTRADS;
KEEP NAME STYPE ID PRINTIT;
RUN;
```

```

X ALLOC F(MAP) DA('SASXXX.MAP.DATA') SHR REU; /* WHERE THE DATA */
/* SET LIVES. */

```

```

DATA DATABASE;
SET MAP.STREETS;
RUN;
```

```

PROC SORT DATA=DATABASE;
BY NAME STYPE;
RUN;
```

```

PROC SORT DATA=STREETS;
BY NAME STYPE;
RUN;
```

```

DATA MERGED; /* KEEP ONLY THE */
MERGE STREETS DATABASE; /* STREETS LISTED */
BY NAME STYPE; /* IN THE STREETS */
IF ID=1; /* DATA SET. */
RUN;
```

```

/* IF YOU ARE INTERESTED IN ONLY INCLUDING SPECIFIC ZIP CODE AREAS,
/* UNCOMMENT THE FOLLOWING SECTION AND FILL IN THE APPROPRIATE
/* ZIP CODES.
/* DATA MERGED;
/* SET MERGED;
/* IF ZIP=XXXXX OR ZIP=XXXXX;
/* RUN;

```

```

/*
| BELOW DETERMINES THE MIN AND MAX OF THE X AND Y VALUES IN THE DATA
| SET MERGED AND RESCALES THE X,Y VALUES TO BE IN THE RANGE OF 0 TO 1
| 100 (WINDOW PERCENTAGE).

```

```

PROC MEANS DATA=MERGED /* FIND THE SMALLEST AND */
NOPRINT /* LARGEST VALUES FOR X AND Y. */
MIN
MAX;
VAR X Y;
OUTPUT OUT=STAT
MIN=XMIN YMIN
MAX=XMAX YMAX;
RUN;
```

```

/* RESCALE THE X AND Y VALUE TO BE IN THE RANGE OF 0 TO 100.*/
DATA MERGED;
DROP XMAX XMIN YMAX YMIN;
IF _N_=1 THEN DO;
SET STAT;
RETAIN XMAX XMIN YMAX YMIN;
END;
SET MERGED;
/* SEE APPENDIX 3. */
%SCALE(X, Y, XMIN, YMIN, XMAX, YMAX, 0, 0, 100, 100);
RUN;
```

```

/*
| BELOW CREATES THE ANNOTATE DATA SET FROM THE RESCALED MERGED DATA
| SET.

```

```

DATA &OUTDATA;
%DCLEANNG; /* DECLARE ANNOTATE VARIABLES. */
%SYSTEM(5,5); /* DEFINE THE ANNOTATE REFERENCE */
SET MERGED; /* SYSTEM (S=WINDOW PERCENTAGE). */

```

```

/*
/* INITIALIZE VARIABLES AND ARRAYS.

```

```

RETAIN ANGDEG 0; /* ANGLE TO PRINT THE STREET'S NAME. */
RETAIN COUNT 0; /* LOOP VARIABLES. */
RETAIN CURRENT; /* CONTAINS THE CURRENT STREET NAME. */
ARRAY XAXIS(2); /* SAVES THE FIRST AND SECOND X,Y */
ARRAY YAXIS(2); /* PAIR FOR EACH STREET. */
RETAIN XAXIS1-XAXIS2;

```

```

RETAIN YAXIS1-YAXIS2;
KEEP FUNCTION X Y XSYS YSYS HSY5 WHEN POSITION SIZE STYLE COLOR
TEXT NAME ANGLE;

/*-----*/
/* CHECK FOR THE BEGINNING OF A NEW STRSET, SET CURRENT TO THE
/* NAME OF THE NEW STREET BEING PROCESSED, STORE THE FIRST X,Y
/* PAIR OF EACH STREET AND SET DRAWTYPE TO 0. THE X,Y PAIR
/* IS USED IN DETERMINING THE ANGLE TO PRINT THE STREET'S NAME.
/*-----*/
IF CURRENT NDT=NAME OR _M_= 1
THEN DO;
DRAWTYPE=0; /* BEGINNING OF A NEW STREET. */
CURRENT=NAME; /* SET CURRENT TO NEW STREET NAME. */

/* STORE THE STREET'S FIRST SET OF COORDINATES.
XAXIS(1)=X; /* USED FOR LABELING.
YAXIS(1)=Y;

COUNT=0; /* COUNTS OBS IN EACH CHARACTER.
END; /* NOT= */

COUNT=COUNT+1;

/*-----*/
/* DETERMINE WHERE TO PLACE THE STREET NAME.
/*-----*/
%SEQUENCE(AFTER);
%LET FAN=XUPCASE("&FANCY");
%IF %INDEX(&FAN,Y)>0
%THEN %DO;
IF PRINTIT=1 AND /* IF PRINTIT=1 THEN OUTPUT NAME.
COUNT=2 /* OTHERWISE NAME WILL NOT BE PRINTED.
THEN DO; /* PRINTIT IS FROM THE INDATA DATA SET.
SAVEX=X; /* SAVE THIS OBSERVATION'S X,Y COORDINATE.
SAVEY=Y;

XAXIS(2)=X; /* SAVE THE SECOND X,Y PAIR
YAXIS(2)=Y; /* IN EACH STREET.

/*-----*/
| THE MACRO FINDANG DETERMINES THE ANGLE TO PRINT THE
| STREET'S NAME.
/*-----*/
ANGDEG=0;
%FINDANG(XAXIS, YAXIS, ANGDEG);

/* PRINT THE STREET'S NAME.
%LABEL(XAXIS(1),YAXIS(1),NAME,BLACK,ANGDEG,0,1.50,XSWISS,0);

X = SAVEX; /* RESTORE THIS OBSERVATION'S X,Y PAIR.
Y = SAVEY;
END;
%END; /* FANCY */
%ELSE %DO;
IF COUNT=1 AND PRINTIT=1
THEN DO;
/* PRINT THE NAME BESIDE THE STREET.
%LABEL(XAXIS(1),YAXIS(1),NAME,BLACK,0,0,1.50,XSWISS,0);
END;
%END;

/*-----*/
/* DRAW THE STREETS.
/*-----*/
%SEQUENCE(BEFDRE);
SIZE=0.5;
COLOR='BLACK';
IF DRAWTYPE=0 THEN FUNCTION='MOVE'; /* DRAWTYPE IS FROM THE
ELSE FUNCTION='DRAW'; /* DATABASE DATA SET.
OUTPUT;
RUN; /* OUTOATA */

/*-----*/
| BELOW, THE DATA SET LOCATION IS CREATED. THIS DATA SET CONTAINS
| ADDRESS COORDINATES.
| VARIABLES IN DATA SET ARE: NAME = STREET NAME
| ADDRESS NUMBER = ADDRESS NUMBER
| X,Y PAIR.
| THIS SECTION IS ONLY EXECUTED IF MAPADD=YES.
/*-----*/

```

```

%LET ADD=XUPCASE("&MAPADD");
%IF %INDEX(&ADD,Y)>0
%THEN %DO;
DATA LOCATION;
SET &INDATA; /* RENAME THE VARIABLES
RENAME STYPE= STY; /* FOR COMPARISON.
RENAME SPD = DIR; /* STREET PREFIX DIRECTION.
RENAME NAME = STREET; /* STREET TYPE.
/* STREET NAME.
RUN;

PROC MEANS DATA = MERGED /* FIND THE NUMBER OF
NOPRINT /* OBSERVATIONS IN MERGED.
N
NMISS;
VAR X;
OUTPUT OUT = STAT
N = N OBS
NMISS = NMOBS;
RUN;

DATA _BULL_; /* N IS EQUAL TO THE
SET STAT; /* NUMBER OF OBSERVATIONS
N=N OBS+NMOBS; /* IN THE MERGED DATA SET.
CALL SYMPUT('NAXOBS',N);
RUN;

DATA MERGED; /* LRIGHADD IS A CHARACTER
LENGTH ADD 6.; /* VARIABLE. CONSEQUENTLY,
RETAIN ADD; /* IN ORDER TO SORT THE
SET MERGED; /* MERGED DATA SET
/* THE INPUT FUNCTION CONVERTS
/* CORRECTLY, SORT BY A
/* A CHARACTER VALUE TO NUMERIC
/* NUMERIC VARIABLE (ADD)
ADD=INPUT(LRIGHADD,5.); /* TO GET PROPER NUMERIC
/* ORDER.
RUN;

PROC SORT DATA=MERGED;
BY NAME STYPE ADD;
RUN;

/*-----*/
| THE DATA STEP BELOW COMPARES EACH OBSERVATION IN LOCATION
| WITH EVERY OBSERVATION IN MERGED. WHEN A MATCH IS FOUND, THE
| X,Y PAIR FOR THE CURRENT OBSERVATION AND THE FOLLOWING
| OBSERVATION ARE SAVED. THE ADDRESS POINT IS INTERPOLATED
| USING THE X,Y PAIRS SAVED IN XNUM,YNUM AND STORED IN THE
| DATA SET LOCATION.
| FYI: IDX - INDEX POINTER.
| FOUND - 0 MEANS THE ADDRESS HAS NOT BEEN FOUND.
| MOREOBS - LOOP VARIABLE, 0=STOP.
/*-----*/
DATA LOCATION;
LENGTH HIADD LOWADD $ 6; /* HIGH AND LOW
RETAIN HIADD LOWADD; /* ADDRESS NUMBERS.
RETAIN FOUND MOREDBS IDX NEXTOBS; /* FLAGS.
RETAIN DY DX; /* CHANGE IN Y & X
RETAIN NEWX NEWY; /* NEW ADDRESS POINT
ARRAY XNUM(2);
ARRAY YNUM(2);
RETAIN XNUM1-XNUM2;
RETAIN YNUM1-YNUM2;
KEEP STREET STY DIR ADDRNUM NEWX NEWY; /* IF YOUR INDATA
/* HAS AN EVENT TYPE
/* VARIABLE (SEE
/* EXAMPLE IN HEADER)
/* PUT IT IN THE
/* KEEP LIST.
DO WHILE (MOREOBS=1);
SET MERGED POINT=IDX;

/* LOCATE THE SEGMENT POINT WHICH CONTAINS THE ADDRESS OF
/* THE STREET PRESENTLY BEING PROCESSED.
IF ( NAME=STREET ) AND ( STYPE=STY ) AND ( SPD=DIR ) AND
( { ADDRNUM=>LRIGHADD } AND ( ADDRNUM<=HLEFTADD ) )
THEN DO;
FOUND = 1;
MOREOBS = 0;
XNUM(1) = X;
YNUM(1) = Y;
LOWADD = INPUT(LRIGHADD,5.);

/* READ THE NEXT OBSERVATION IN THE DATA SET MERGED.
NEXTOBS=IDX + 1;
SET MERGED POINT=NEXTOBS;
XNUM(2)=X;
YNUM(2)=Y;
HIADD=INPUT(HLEFTADD,5.);

/* INTERPOLATE ADDRESS POINT.

```

Appendix 3

```

DY=YNUM(2)-YNUM(1);          /* CHANGE IN X      */
DX=XNUM(2)-XNUM(1);          /* CHANGE IN Y      */
RATIO=((ADDRNUM-LOWADD) / (HIADD-LOWADD));
NEWX=XNUM(1)+RATIO*DX;        /* (NEWX,NEWY)=ADDRESS */
NEWY=YNUM(1)+RATIO*DY;
END; /* FOUND SEGMENT */

ELSE DO;
  IDX=IDX+1;
  IF IDX>MAXOBS THEN MOREOBS=0; /* END OF FILE REACHED */
  END;
END; /* WHILE */

/* OUTPUT THE ADDRESS POINT. */
IF FOUND=1
  THEN OUTPUT;

ELSE PUT 'NOTE: COULD NOT FIND ADDRESS FOR ' ADDRNUM ' STREET;
RUN;

DATA LOCATION;
SET LOCATION;
RENAME NEWX=X;
RENAME NEWY=Y;
RUN;

/* IF YOU HAVE AN EVENT TYPE VARIABLE IN YOUR INDATA, YOU MAY WISH */
/* TO CHANGE THE SYMBOL USED TO MARK THE ADDRESS TO CORRESPOND TO */
/* THE VALUE OF THE EVENT VARIABLE. */
DATA LOCATION;
SET LOCATION;
%DCLEARND;
%SYSTEM(5,5,5);
KEEP FUNCTION X Y XSYS YSYS HSYS WHEN POSITION SIZE STYLE COLOR
TEXT STREET;
%LABEL(X,Y,'M',RED ,0.,0.,3.50,SPECIAL,5); /* M='STAR' */
RUN;

DATA %OUTDATA;
SET %OUTDATA LOCATION;
RUN;

%END; /* MAPADD */

-----
| OUTPUT THE MAP USING PROC GSLIDE. |
-----
PROC GSLIDE ANNOTATE=%OUTDATA;
RUN;

RUN;
%MEND MAPCITY;

```

```

-----
| THE FOLLOWING MACROS ARE USED IN THE MACRO MAPCITY. |
-----
/*
| FINDANG FINDS THE ANGLE USED TO PRINT THE STREET'S NAME. |
-----
%MACRO FINDANG(XAXIS, YAXIS, ANGDEG);
DO;
  RETAIN ANGRAD;          /* ANGLE FOUND BETWEEN TWO PDINTS.*/
  RETAIN CH_X  CH_Y;      /* CHANGE IN X, CHANGE IN Y.      */
  RETAIN R2D    57.29583; /* 180 / PI                        */

  /* DETERMINE THE CHANGE IN X AND Y. */
  CH_X=XAXIS(2)-XAXIS(1);
  CH_Y=YAXIS(2)-YAXIS(1);

  /* FIND ANGLE -90. (NAME IS WRITTEN FROM TOP TO BOTTOM) */
  IF (CH_X=0E0) THEN ANGRAD=-1.5707963;

  /* FIND ANGLE 0. */
  ELSE IF (CH_Y=0E0) THEN ANGRAD=0;

  ELSE DO;
    /*
    | ARCTANGENT (ATAN FUNCTION) IS THE INVERSE |
    | FUNCTION TO THE TANGENT. |
    -----
    ANGRAD=ATAN(CH_Y/CH_X);
    /*
    | BELOW IS FOR BETTER READABILITY. |
    | READ STREET NAME FROM TOP TO BOTTOM. |
    -----
    IF (ANGRAD>1.396263402) /* 80 DEGREES */
      THEN ANGRAD=-ANGRAD;

    END;

    /* CHANGE THE ANGLE INTO DEGREES. */
    ANGDEG=ANGRAD/R2D;
  END;
%MEND;

/*
| SCALES THE VALUE PTX,PTY TO BE BETWEEN 'VXMIN,VYMIN' AND |
| 'VXMAX,VYMAX'. |
-----
%MACRO SCALE( PTX,PTY, XMIN,YMIN, XMAX,YMAX, VXMIN,VYMIN, VXMAX,VYMAX );
DO;
  %LET F1 = ( (&VXMAX-&VXMIN) / (&XMAX-&XMIN) );
  %LET F2 = ( (&VYMAX-&VYMIN) / (&YMAX-&YMIN) );
  X = &F1 * (&PTX-&XMIN);
  Y = &F2 * (&PTY-&YMIN);
END;
%MEND SCALE;

/*
| THE FOLLOWING CREATES A NORTH MARKER FOR THE MAP. |
| MX,MY = THE X,Y COORDINATE WHERE YOU WISH TO |
| PLACE THE MARKER. |
-----
%MACRO COMPROSE( MX, MY, NCOLOR);
DO;
  /* SAVE XSYS, YSYS AND HSYS */
  XSYS=XSYS;
  YSYS=YSYS;
  HSYS=HSYS;

  /* MOVE TO LOCATION WHERE MARKER WILL BE DRAWN. */
  %MOVE ( &MX, &MY );

  XSYS="9"; /* '9': SCREEN PERCENTAGE. */
  YSYS="9"; /* '9' TO 100% OF PAGE WINDOW. */
  HSYS="3";

  /* 'SLICE': SPECIFIES TO DRAW AN ARC OR SECTION*/
  /* OF A FILLABLE CIRCLE WITH THE CENTER AT */
  /* (X1,Y1). */
  %SLICE( +0, +0, 0,360,2,&NCOLOR,SOLID, 1);

  %DO DEGS=0 %TO 315 %BY 45;
  /* CREATES THE NORTH MARKER. */

```



```

%IF &DEGS=45 %THEN %DO;
/* 'PIEXY': SPECIFIES TO LOCATE A POINT */
/* ALONG A RADII OF MULTIPLIER LENGTH BASED */
/* ON THE PREVIOUSLY DRAWN SLICE. */
%PIEXY(&DEGS, .9);
/* 'POLY': SPECIFIES TO BEGIN DRAWING A */
/* POLYGON. */
%POLY(.,.,&MCOLOR, SOLID, 1);
%PIEXY(&DEGS+45, 5.5);
/* 'POLYCONT': SPECIFIES TO CONTINUE */
/* DRAWING THE POLYGON. */
%POLYCONT(-999,-999,&MCOLOR);
%PIEXY(&DEGS+90, .9);
%POLYCONT(-999,-999,&MCOLOR);
%END;

/* OTHERWISE, CREATE THE OTHER MARKERS. */
%ELSE %DO;
%PIEXY(&DEGS, .9);
%POLY(.,.,&MCOLOR, SOLID, 1);
%PIEXY(&DEGS+45, 3);
%POLYCONT(-999,-999,&MCOLOR);
%PIEXY(&DEGS+90, .9);
%POLYCONT(-999,-999,&MCOLOR);
%END;

/* RESTORE XSYS, YSYS AND HSYS. */
XSYS=SYSX; YSYS=SYSY; HSYS=SYSH;

/* LABEL THE NORTH MARKER. */
%LABEL(&MX, &MY+15, 'N', &MCOLOR, 0, 0, 3, SIMPLEX, 5);
%MOVE( &MX, &MY );
END;
%MEMD COMPROSE;

```