

UNDERSTANDING THE ANNOTATE FACILITY REFERENCE SYSTEMS

Anthony L. Friebel
SAS Institute Inc.

The reference systems form a unique framework for data handling within the ANNOTATE Facility. This framework permits a wider degree of flexibility when communicating graphic requests of the system. No longer are you forced to convey your message only in grid cell measures. Percentage measurements, relative measurements, and, for the first time, actual data values can be used in referencing coordinates on a graph. These unit specifications may be intermixed in any of the ANNOTATE Facility functions.

I. ABSOLUTE SYSTEMS

A. SCREEN VALUE / SCREEN PERCENTAGE

Users of the NOTE statement are already familiar with the SCREEN VALUE ('4') system. The display area is divided into grid cells as measured by the GOPTIONS HPOS= and VPOS= parameters. SCREEN PERCENTAGE ('5') defines this area in a more constant 0 to 100 field. Because the SCREEN VALUE system depends entirely on the HPOS= and VPOS= values to determine screen size, it is a more volatile coordinate system as parameters change from device to device. By always specifying the range as 0 to 100, the SCREEN PERCENT system helps to guarantee device independence of ANNOTATE generated graphics.

For example, on an IBM3279 (default HPOS=80, VPOS=32) a coordinate X=40 in SCREEN VALUE is equivalent to X=50 in SCREEN PERCENTAGE units and produces a location in the center of the display area. See Figure 1. If we now move to device XYZZY where the defaults are HPOS=160 and VPOS=80, the X=40 SCREEN VALUE coordinate "changes position" while the SCREEN PERCENTAGE location maintains its position in the center of the screen, illustrated in Figure 2.

Figure 1.
IBM3279

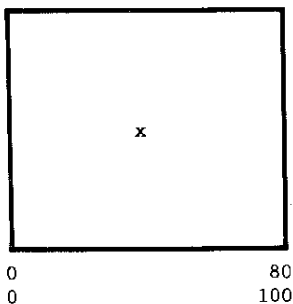
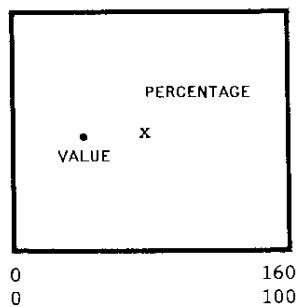


Figure 2.
XYZZY



Example 1.

The NOTE statement shown produces a box on the screen. The data set CELLS performs the identical function and illustrates the conversion of the NOTE statement format into the equivalent ANNOTATE data step code.

Old NOTE format :

```
PROC GSLIDE;
NOTE .C=GREEN .D=( 30, 10, 50, 10,
RUN;                               50, 20, 30, 20, 30, 10);
```

ANNOTATE format :

```
DATA CELLS;
LENGTH FUNCTION COLOR $ 8;
XSYS = '4'; YSYS = '4';
COLOR='GREEN';
FUNCTION='MOVE'; X=30; Y=10; OUTPUT;
FUNCTION='DRAW'; X=50; Y=10; OUTPUT;
FUNCTION='DRAW'; X=50; Y=20; OUTPUT;
FUNCTION='DRAW'; X=30; Y=20; OUTPUT;
FUNCTION='DRAW'; X=30; Y=10; OUTPUT;
RUN;
PROC GSLIDE ANNOTATE=CELLS;
RUN;
```

The code below converts the above value specifications to the equivalent percentage values.

```
DATA CELL2PCT;
SET CELLS;
DROP HPOS VPOS;
RETAIN HPOS 80; /* original device */
RETAIN VPOS 32; /* original device */

IF X ^= . THEN IF XSYS='4' THEN DO;
X = 100*( X / HPOS );
XSYS = '3';
END;

IF Y ^= . THEN IF YSYS='4' THEN DO;
Y = 100*( Y / VPOS );
YSYS = '3';
END;

IF SIZE ^= . THEN IF HSYS='4' THEN DO;
SIZE = 100*( SIZE / VPOS );
HSYS = '3';
END;

RUN;
PROC GSLIDE ANNOTATE=CELL2PCT;
RUN;
```

B. WINDOW VALUE / WINDOW PERCENTAGE

Window systems and the screen systems are similarly defined with one exception. They differ in their handling of TITLE and FOOTNOTE statements. The screen systems ignore the space occupied by title or footnote text and define the entire screen area. Window systems, on the other hand, respect this space and restrict their activity to the area enclosed by the TITLES and FOOTNOTES. If no TITLES or FOOTNOTES are in use, the window systems and the screen systems are identical.

The SAS/GRAPH procedures work in a "window" environment. No graphics are generated which would overwrite the TITLES or FOOTNOTES. Thus, the window systems are ideal for use when creating custom graphics procedures.

Figures 3 and 4 below illustrate the difference between screen and window reference systems.

Figure 3.
SCREEN SYSTEMS

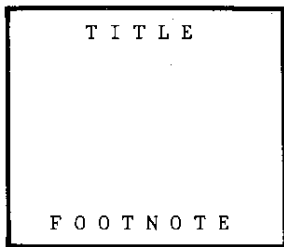
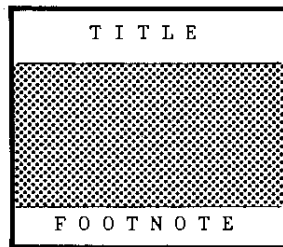


Figure 4.
WINDOW SYSTEMS



Window systems use only the area shaded in Figure 4 above. The title and footnote area is 'protected'. The usable area for graphics is defined as the entire screen less the amount required by TITLE and FOOTNOTE statements.

C. DATA VALUE / DATA PERCENTAGE

These systems use actual data values, or a percentage of the data area, to specify their coordinates. Currently PROC Gplot, GMAP, G3D, GCONTOUR and GCHART support these systems, as they all have recognizable data ranges. All but GMAP produce visual axes to delineate this range. For procedures like GSLIDE or GANNO which have no actual axes, the ANNOTATE Facility assumes the range to be 0 to 100 in both directions. You will recognize these values as the SCREEN PERCENTAGE system discussed earlier.

For an example using the DATA systems, let us turn to PROC Gplot. In Example 2, we have constructed a vertical axis from 1 to 10 and a horizontal axis from 5 to 30. The curve generated has a minimum at X=10, Y=3. Its maximum is X=25, Y=8. We want to place descriptive labels at these two points, so we construct an ANNOTATE data set using the DATA VALUE ('2') reference system for both the horizontal (X) and the vertical (Y) axes. Notice that the X and Y coordinates specified in the data set are the actual values of the points we want to annotate. The first two observations perform the labelling functions.

If we also wanted to place a label centered directly above the axis area, we could use the data percentage ('1') reference system. To center horizontally, we measure half of the span, or 50 percent. We specify X=50; for this value. Vertically we want the label at the top of the frame, i.e. 100 percent of the data area, so we specify Y=100; to complete the coordinate pair. (Because this upper limit is also the data value Y=10, we could have also used the DATA VALUE system, specified YSYS='2'; Y=10; and achieved equivalent results. Remember, systems for X and Y are independent of one another. We may choose whichever system best fits our needs.)

Example 2.

```

DATA POINTS;
  INPUT X Y;
  CARDS;
    5 5
   10 3
   15 4
   20 7
   25 8
   30 6
  RUN;

DATA DESCRIPT;
  LENGTH TEXT $ 8;
  XSYS='2'; YSYS='2';
  TEXT='MINIMUM '; X= 10; Y= 3; OUTPUT;
  TEXT='MAXIMUM '; X= 25; Y= 8; OUTPUT;
  XSYS='1'; YSYS='1';
  TEXT='CENTERED'; X= 50; Y=100; OUTPUT;
RUN;

PROC Gplot      DATA=POINTS;
  AXIS1         ORDER = 5 TO 30 BY 5
                OFFSET= (0,0)
                MAJOR = NONE
                MINOR = NONE;

  AXIS2         ORDER= 1 TO 10
                OFFSET= (0,0)
                MAJOR = NONE
                MINOR = NONE;

  PLOT Y*X /    HAXIS= AXIS1
                VAXIS= AXIS2
                ANNOTATE=DESCRIPT;

  SYMBOL I=JOIN;
RUN;

```

In Example 2, we knew the minimum and maximum values of the data and chose the coordinates accordingly. We could as easily have calculated the min/max points and generated these points automatically. Example 3 illustrates this by using PROC MEANS to determine the data minima and maxima and data step code to generate the labels. The results are the same as in Example 2. Only the methodology has changed. This version will always reflect changing data values.

Example 3.

```
PROC MEANS DATA=POINTS NOPRINT ;
VAR X Y;
OUTPUT      OUT=STAT
            N =NOBS
            MIN=XMIN  YMIN
            MAX=XMAX  YMAX;
RUN;
```

```
DATA DESCRIPT;
LENGTH TEXT $ 8;
RETAIN  XMAX XMIN YMAX YMIN NOBS;
DROP    XMAX XMIN YMAX YMIN NOBS;

IF _N_ = 1 THEN SET STAT;
SET POINTS;
IF _N_ < NOBS
THEN DO;
  XSYS='2'; YSYS='2';
  SELECT ( Y );
    WHEN( YMIN ) DO;
      TEXT='MINIMUM';
      OUTPUT;
    END;
    WHEN ( YMAX ) DO;
      TEXT='MAXIMUM';
      OUTPUT;
    END;
    OTHERWISE ;
  END; /* SELECT */
END;
ELSE DO;
  XSYS='1'; YSYS='1';
  X= 50; Y=100;
  TEXT='CENTERED';
  OUTPUT;
END;
RUN;
```

```
PROC GPLOT      DATA=POINTS;
  AXIS1         ORDER = 5 TO 30 BY 5
                OFFSET= (0,0)
                MAJOR = NONE
                MINOR = NONE;

  AXIS2         ORDER= 1 TO 10
                OFFSET= (0,0)
                MAJOR = NONE
                MINOR = NONE;

  PLOT Y*X /    HAXIS= AXIS1
                VAXIS= AXIS2
                ANNOTATE=DESCRIPT;
  SYMBOL I=JOIN;
RUN;
```

II. RELATIVE SYSTEMS

Each of the six reference systems previously described has a counterpart in the relative reference system scheme. In a relative coordinate system, a distance is measured from the previous position, rather than always being measured from a fixed position, or origin, as in the absolute systems.

Example 4. Old NOTE format :

```
PROC GSLIDE;
  NOTE .C=GREEN .D=( 30, 10, +20, +0,
                    +0, +10, -20, +0, +0, -10);
RUN;
```

ANNOTATE format :

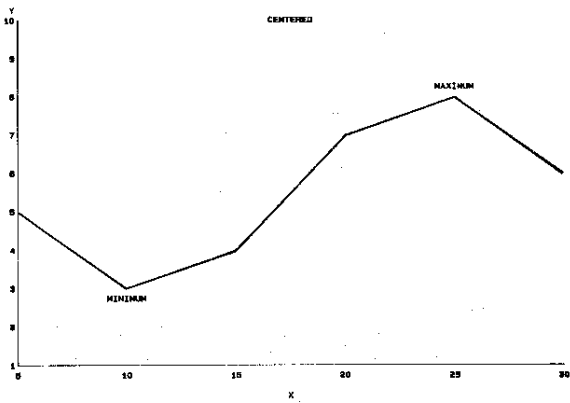
```
DATA RELCELLS;
LENGTH FUNCTION COLOR $ 8;
XSYS = 'A'; YSYS = 'A';
COLOR='GREEN';
FUNCTION='MOVE'; X= 30; Y= 10; OUTPUT;
FUNCTION='DRAW'; X= 20; Y= 0; OUTPUT;
FUNCTION='DRAW'; X= 0; Y= 10; OUTPUT;
FUNCTION='DRAW'; X=-20; Y= 0; OUTPUT;
FUNCTION='DRAW'; X= 0; Y=-10; OUTPUT;
RUN;
PROC GSLIDE ANNOTATE=RELCELLS;
RUN;
```

Because the ANNOTATE Facility begins coordinates at (0,0) in all systems, ABSOLUTE X=30, Y=10 is equivalent to RELATIVE X=30, Y=10 as a first observation value in the SCREEN VALUE systems. Notice that this is not true for the NOTE statement shown above. Direct translation for the NOTE would be as follows :

```
DATA THENOTE;
LENGTH FUNCTION COLOR $ 8;
COLOR='GREEN';
XSYS = '4'; YSYS = '4';
FUNCTION='MOVE'; X= 30; Y= 10; OUTPUT;

XSYS = 'A'; YSYS = 'A';
FUNCTION='DRAW'; X= 20; Y= 0; OUTPUT;
FUNCTION='DRAW'; X= 0; Y= 10; OUTPUT;
FUNCTION='DRAW'; X=-20; Y= 0; OUTPUT;
FUNCTION='DRAW'; X= 0; Y=-10; OUTPUT;
RUN;
PROC GSLIDE ANNOTATE=THENOTE;
RUN;
```

ANNOTATE REFERENCE SYSTEMS

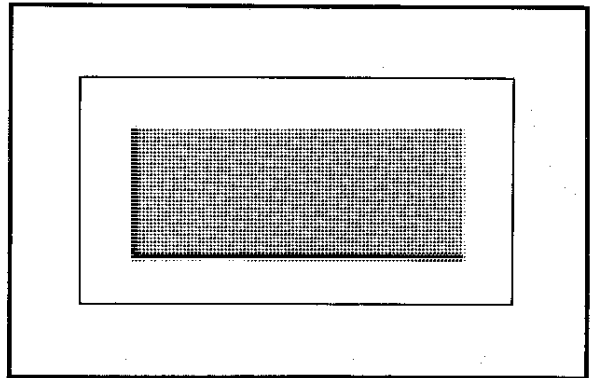


The following is a list of annotate support macros and their parameter lists. The source for these macros is included in the SAS sample library on the distribution tape.

```

%DCLANNO;
%COMMENT( TXT );
%SEQUENCE( SEQ );
%SYSTEM( XS, YS, HS );
%MOVE( X1, Y1 );
%DRAW( X1, Y1, COLOR, LINTYP, WIDTH );
%LABEL( X1, Y1, TXT, COLOR, ANGLE, ROTATE, HEIGHT, FONT, POS );
%RECT( X1, Y1, X2, Y2, COLOR, LINTYP, WIDTH );
%BAR( X1, Y1, X2, Y2, COLOR, BARTYP, PATTERN );
%CIRCLE( X1, Y1, RAD, COLOR );
%SLICE( X1, Y1, ANGLE, ROTATION, RADIUS, COLOR, PATTERN, LTYP );
%PIEXY( ANGLE, MULTIPLIER );
%POLY( X1, Y1, COLOR, PATTERN, LINTYP );
%POLYCONT( X1, Y1, COLOR );
%SCALE( PTX, PTY, XMIN, YMIN, XMAX, YMAX, VXMIN, VYMIN, VXMAX, VYMAX );
%PUSH;
%POP;
%SWAP;
%TXT2CNTL;
%CNTL2TXT;
%DRAW2TXT( COLOR, LINTYP, WIDTH );
    
```

Under CMS the following names are changed (sequence, when), (polycont, polycon), (txt2cntl, txt2ctl), (cntl2txt, ctl2txt), (draw2txt, drw2txt)

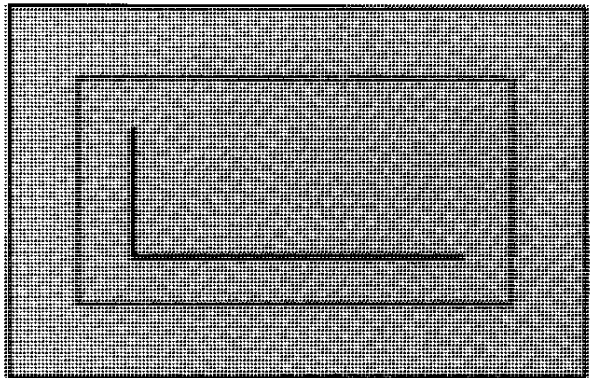


SYSTEM 1
data percentage

SYSTEM 2
data value

SYSTEM 7
data relative percentage

SYSTEM 8
data relative value

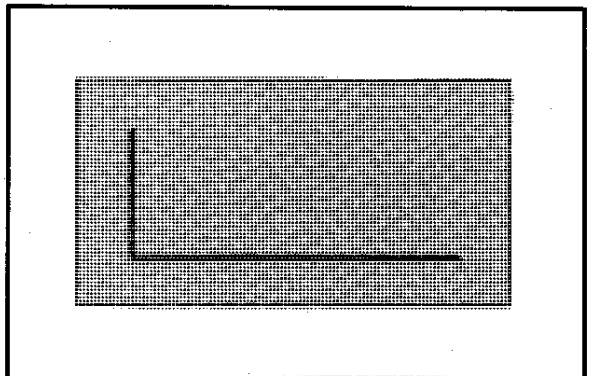


SYSTEM 3
screen percentage

SYSTEM 4
screen value

SYSTEM 9
screen relative percentage

SYSTEM A
screen relative value



SYSTEM 5
window percentage

SYSTEM 6
window value

SYSTEM B
window relative percentage

SYSTEM C
window relative value